

Identifikasi Kendaraan Beroda Menggunakan Algoritma YOLOv5

Michael

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
Jalan Diponegoro No. 52 - 60, Kel. Salatiga, Kec. Sidorejo, Kota Salatiga, Prov. Jawa Tengah
Email: 672020033@student.uksw.edu

Abstract. Wheeled Vehicle Identification Using YOLOv5 Algorithm. *The importance of traffic density measurement in road planning has led to efforts in automation using object detection algorithms, particularly YOLO (You Only Look Once), which are replacing error-prone and time-consuming manual processes. However, challenges arise in dense traffic conditions, posing a challenge to vehicle detection accuracy. This research aims to compare the performance of vehicle detection between two YOLO approaches: multi-view layer detection and conventional detection, focusing on YOLOv5n, YOLOv5s, and YOLOv5m. The literature review encompasses Computer Vision, YOLO implementation, and related research to provide conceptual context. The research method details the steps of vehicle identification using YOLOv5, and the evaluation includes the performance of various YOLO variants and multi-view detection approaches. Thus, this study is expected to gain deeper insights into building an effective model and facilitating the selection of a suitable YOLO model for vehicle detection.*

Keywords: *traffic density, object detection, YOLO (You Only Look Once), vehicle detection*

Abstrak. *Pentingnya pengukuran kepadatan lalu lintas dalam perencanaan jalan memunculkan upaya otomatisasi menggunakan algoritma deteksi objek, terutama YOLO (You Only Look Once), yang menggantikan proses manual yang rawan kesalahan dan memakan waktu. Meski demikian, tantangan muncul saat lalu lintas padat, menantang akurasi deteksi kendaraan. Dalam rangka mengatasi kendala ini, penelitian ini bertujuan membandingkan performa deteksi kendaraan antara dua pendekatan YOLO: deteksi lapisan multi-view dan deteksi konvensional, dengan fokus pada YOLOv5n, YOLOv5s, YOLOv5m, dan YOLOv5l. Kajian literatur melibatkan bidang Computer Vision, implementasi YOLO, dan riset terkait guna memberikan konteks konseptual. Metode penelitian mendetailkan langkah-langkah identifikasi kendaraan dengan YOLOv5, dan evaluasi hasilnya mencakup performa berbagai varian YOLO dan pendekatan deteksi multi-view. Dengan demikian, melalui penelitian ini, diharapkan akan diperoleh wawasan yang lebih mendalam untuk membangun model yang efektif serta memudahkan pemilihan model YOLO yang sesuai untuk deteksi kendaraan.*

Kata Kunci: *kepadatan lalu lintas, deteksi objek, YOLO (You Only Look Once), deteksi kendaraan*

1. Pendahuluan

Pengukuran kepadatan lalu lintas merupakan aspek krusial dalam perencanaan jalan, yang mencerminkan jumlah kendaraan per satuan panjang jalan tertentu. Parameter ini memiliki dampak signifikan dalam menentukan geometri jalan, ketebalan perkerasan, serta dalam evaluasi perkiraan ekonomi sektor jalan. Namun pertumbuhan kendaraan yang tinggi tidak dibarengi dengan pembangunan jalan yang memadai, sehingga banyak kendaraan dan jalan yang rusak menyebabkan kemacetan lalu lintas yang merugikan masyarakat. Oleh karena itu, perlu dibangun atau diperlebar jalan guna mengurangi kemacetan yang terjadi [6].

Dalam kondisi lalu lintas padat, kendala terkait akurasi deteksi kendaraan muncul karena objek kendaraan yang berdekatan dapat menyulitkan jaringan deteksi. Berbagai detektor kendaraan, termasuk yang menggunakan *Convolutional Neural Network* (CNN) dan *Region Based Convolutional Neural Network* (R-CNN), telah dikembangkan untuk mengatasi masalah ini. Meskipun berhasil, R-CNN memiliki keterbatasan kecepatan, sedangkan metode deteksi

objek yakni *You Only Look Once* (YOLO), seperti varian YOLOv5l, mendekati deteksi sebagai masalah regresi langsung, membagi gambar menjadi sel atau *grid* untuk menghitung probabilitas, koordinat, dan kelas objek secara simultan. Tantangan ini memunculkan solusi inovatif, seperti pelabelan kelas multiperspektif yang dapat meningkatkan kinerja jaringan deteksi [4].

Penelitian ini mengambil pendekatan perbandingan kinerja deteksi kendaraan antara dua metode YOLO, yakni deteksi lapisan *multi-view* dan deteksi konvensional. Fokus utamanya adalah pada varian YOLO, seperti YOLOv5n, YOLOv5s, YOLOv5m, dan YOLOv5l [1]. Dengan menggunakan pendeteksian objek metode YOLO pada suatu sistem dapat membantu mengklasifikasi setiap jenis kendaraan yang melintas pada jalan raya secara *real-time* pada rekaman video [10].

Melalui penelitian ini, diharapkan bisa memberikan adanya pemahaman yang lebih mendalam untuk mengatasi kendala deteksi kendaraan pada kondisi lalu lintas padat, serta memberikan panduan praktis untuk membangun model yang efektif menggunakan varian YOLO.

2. Tinjauan Pustaka

Pada penelitian H. Dawami, E. Rachmawati, dan M. D. Sulistiyo, karena keunggulannya dalam akurasi, kemampuan mendeteksi objek kecil, dan kecepatan proses, YOLOv5 adalah pilihan yang baik untuk deteksi masker wajah. Selain itu ada bukti bahwa kinerja model lebih baik dengan menggunakan *dataset* pemindaian muka yang diperluas [2].

Penelitian mengenai “Deteksi Penggunaan *Safety Helmet* Menggunakan YOLOv5” yang dilakukan oleh N. A. K. D. Pasongko, A. Khairunnisa, dan S. Aras, melakukan pembagian *dataset* yang terdiri atas 70% data pelatihan (*training data*), 20% data validasi (*validation data*), dan 10% data pengujian (*test data*). Langkah terakhir dalam membuat kumpulan data adalah mengubah *dataset* yang telah dibuat menjadi format yang sesuai untuk proses pelatihan di YOLOv5 [3]. Penelitian C. Dewi, dkk dengan judul “YOLOv5 *Series Algorithm for Road Marking Sign Identification*”. Makalah penelitian ini memberikan rangkuman rangkaian algoritma YOLO5 untuk identifikasi rambu marka jalan, yang meliputi YOLOv5s, YOLOv5m, YOLOv5n, YOLOv5l, dan YOLOv5x. Studi ini mengeksplorasi berbagai macam detektor objek kontemporer, seperti yang digunakan untuk menentukan lokasi rambu marka jalan [4].

Observasi pada penelitian yang berjudul “Deteksi Kendaraan Secara Real-Time Menggunakan Metode YOLO Berbasis Android”, memakai analisa yakni jumlah *dataset* sebanyak 200, empat kelas, 10 *batch*, serta 200 *epoch*. Adapun proses pelatihan yakni pada 4.000 *step*, serta penyimpanan *checkpoint* kepada *protobuf file* yakni pada *step* 800, kemudian 1.000, 1.200, 1.400, 1.600, 1.800, 2.000, 3.000, serta 4.000. *Bounding box* berhasil mendeteksi serta mengklasifikasi objek secara tepat. Penelitian ini memakai perangkat *smartphone* Xiaomi Redmi 4X pada resolusi video yakni 768x432 piksel [8].

Menurut (Ju, Luo dan Wang 2020) dalam penelitiannya yang berjudul "An improved YOLO V3 for small vehicles detection in aerial images". Hasil eksperimen pada *dataset VEDAI* mendapat peningkatan dengan menggunakan YOLO V3 dengan mencapai nilai 70,3% [9]. Dengan menggunakan algoritma yang sama dari penelitian terdahulu, penulis mencoba untuk melakukan penelitian terkait implementasi algoritma YOLOv5 dengan melakukan peningkatan yaitu dengan menambahkan data baru yang berasal dari internet lalu data tersebut diproses menggunakan data *augmentation*. Data *augmentation* sendiri adalah proses untuk memperbanyak data dengan mengubah sudut pandang dari suatu gambar, seperti *rotate* atau *flip*, proses ini digunakan untuk melatih komputer agar deteksi gambar lebih akurat.

3. Metodologi

3.1. Metode

Metode observasi pada lima tahap dapat dilihat di Gambar 1. *Output* dari setiap tahap merupakan *input* bagi tahapan berikutnya. Metode ini memiliki tahapan yakni: analisis, pengumpulan data, implementasi, pengujian, dan pembuatan laporan.

$$x = \frac{(x_1 + x_2)}{2} X dw \quad (2)$$

$$dh = 1/H \quad (3)$$

$$y = \frac{(y_1 + y_2)}{2} X dw \quad (4)$$

$$w = (x_1 + x_2) X dw \quad (5)$$

$$h = (y_1 + y_2) X dh \quad (6)$$

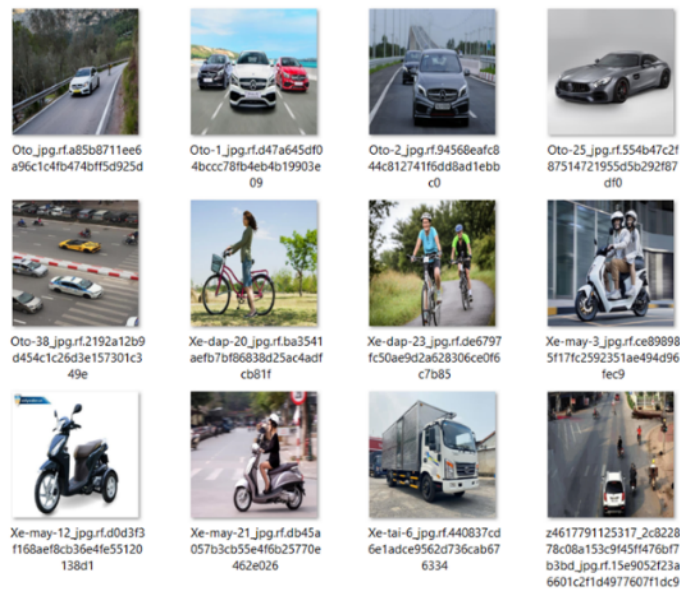
h mengindikasikan tinggi dari gambar, dh merujuk pada tinggi mutlak gambar, w merupakan lebar dari gambar, dan dw mewakili lebar mutlak gambar.

$$\begin{aligned} \Pr(Class_i|Object) * \Pr(Object) * IOU_{ped}^{truth} \\ = \Pr(Class_i) * IOU_{ped}^{truth} \end{aligned} \quad (7)$$

Komponen pertama di sisi kiri Persamaan 7 adalah hasil prediksi yang diperoleh dari setiap *grid*, sementara kedua komponen terakhir adalah hasil prediksi yang masing-masing diperoleh dari setiap kotak (*box*). Ketika ketiga komponen ini digabungkan, mereka membentuk skor yang mengindikasikan bahwa setiap kotak mungkin berisi objek dari kategori yang berbeda. Oleh karena itu, total jumlah nilai prediksi yang dihasilkan oleh jaringan konvolusi adalah $S \times S \times (B \times 5 + C)$, dimana S mewakili jumlah *grid*, B adalah jumlah kotak yang dihasilkan untuk setiap *grid*, dan C adalah jumlah kategori yang mungkin. Tahap pasca pemrosesan: pada tahap ini, kotak-kotak yang diperoleh melalui prediksi akan disaring menggunakan algoritma *Non-Maximum Suppression* (NMS) untuk menghasilkan hasil deteksi akhir [5].

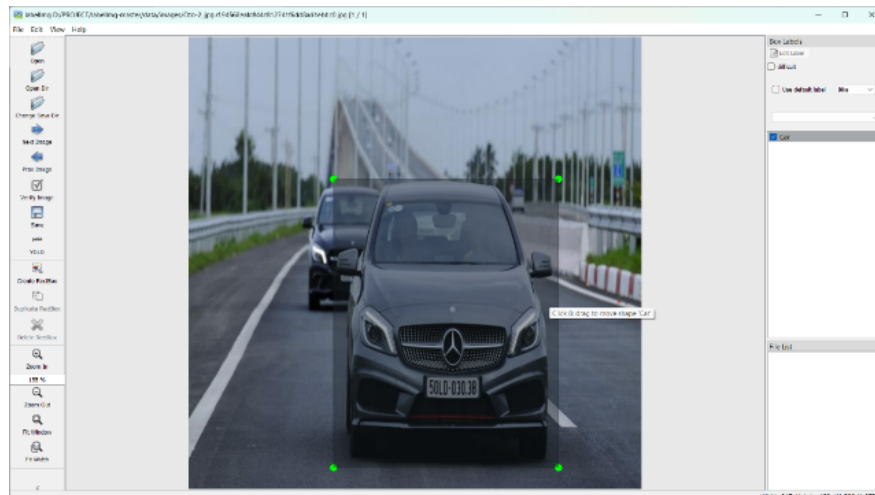
4. Hasil dan Diskusi

Proses pengumpulan data dari internet untuk memperbanyak *dataset* melalui Roboflow melibatkan sejumlah langkah. Informasi lebih lanjut tentang hasil data yang diperoleh dari internet dapat ditemukan dalam Gambar 3. Tahap ini memungkinkan penambahan variasi dan kompleksitas pada *dataset*, meningkatkan kemampuan model untuk mengenali objek dengan lebih baik. Hal ini juga memberikan kesempatan untuk menangkap variasi yang mungkin tidak tercakup dalam *dataset* awal, menghasilkan model yang lebih handal dan responsif terhadap situasi dunia nyata.



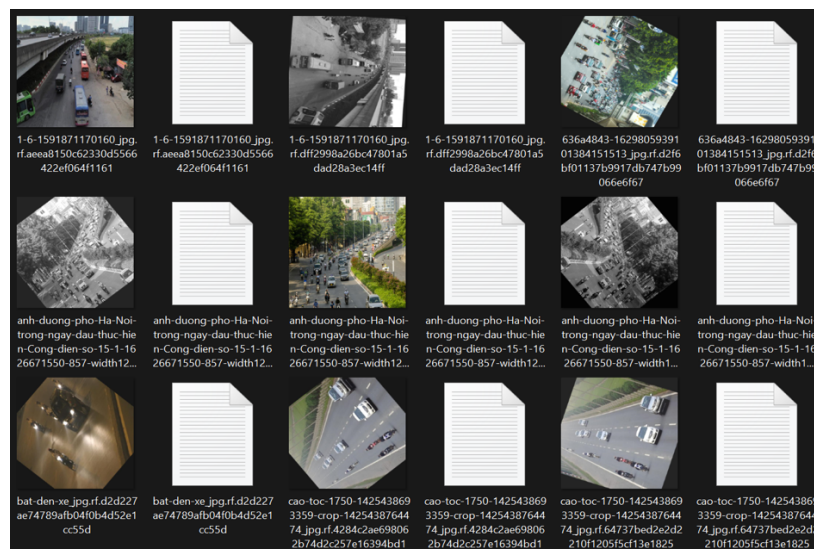
Gambar 3. Foto Data dari Internet

Proses pelabelan objek terbagi jadi lima kelas utama, yakni sepeda, bus, mobil, sepeda motor, serta truk. Objek yang mendapatkan label hanya terlihat pada bagian belakang sesuai dengan foto yang diambil dari internet. Pada tahap pelabelan mobil, fokusnya dimulai pada atap hingga roda. Sementara pada sepeda motor, pelabelan dimulai dari bagian kepala hingga roda. Proses pelabelan truk mencakup area pada bagian atas bak hingga roda truk. Untuk bus, pelabelan dimulai pada atap hingga roda. Terakhir, pelabelan pada sepeda dimulai pada atas stang hingga roda belakang sepeda. Detail skenario pelabelan divisualisasikan dalam Gambar 4. Setelah selesai melakukan anotasi, hasilnya akan tampak seperti pada Gambar 5 yang memuat file gambar serta file .txt.



Gambar 4. Proses Anotasi Dataset Menggunakan LabelImg

Proses anotasi citra dengan menggunakan aplikasi LabelImg mencakup adopsi format anotasi YOLO. Setelah melalui proses anotasi, hasilnya berupa data yang memberikan informasi mengenai lokasi kotak pembatas serta label objek pada *file* .txt. Dalam struktur *file* .txt tersebut, setiap baris adanya format `<object-class> <x_center> <y_center> <width> <height>`. Di mana `<object-class>` yakni bilangan bulat yang merepresentasikan kelas objek, `<x_center>` dan juga `<y_center>` menyatakan koordinat pusat dari kotak pembatas, sementara `<width>` dan juga `<height>` yakni nilai float yang berkaitan dengan dimensi gambar.



Gambar 5. Gambar yang Sudah Diberi Label

Hasil dari proses anotasi ini, yang tersimpan dalam file .txt, memberikan gambaran jelas mengenai letak dan kelas objek. Contoh hasil dapat dilihat pada Gambar 6. Proses anotasi dengan format YOLO ini memberikan data yang diperlukan untuk melatih model deteksi objek dengan lebih akurat dan spesifik.

```
0 0.44453125 0.709375 0.3171875 0.58125
0 0.73046875 0.65625 0.2109375 0.45703125
```

Gambar 6. Isi Salah Satu File .txt

Konfigurasi model YOLOv5 melibatkan pemilihan arsitektur yang sesuai dengan kebutuhan spesifik, serta penyesuaian parameter yang memengaruhi kinerja deteksi objek. Dalam implementasi YOLOv5, dapat dipilih dari berbagai varian model, seperti YOLOv5n, YOLOv5s, YOLOv5m, atau YOLOv5l, yang masing-masing memiliki karakteristik dan kompleksitas yang berbeda.

Penentuan jumlah kelas objek yang akan dideteksi merupakan langkah penting dalam konfigurasi model, karena ini mempengaruhi *output* akhir dari model. Selain itu, ukuran gambar *input*, tingkat pembelajaran (*learning rate*), dan jumlah *epoch* dalam proses *training* juga perlu disesuaikan sesuai dengan karakteristik *dataset* dan kebutuhan deteksi objek yang diinginkan.

Tingkat keberhasilan model YOLOv5 juga dapat ditingkatkan dengan menyesuaikan parameter seperti ukuran batch (*batch size*), momentum, dan berbagai parameter *training* lainnya, untuk contoh konfigurasi dapat dilihat pada Tabel 2. Selama proses *fine-tuning*, penting untuk memonitor metrik *training* seperti *loss* dan akurasi untuk menyesuaikan parameter secara iteratif dan meningkatkan performa model.

Tabel 2. Contoh Konfigurasi pada *Weight* YOLOv5l

Jenis Konfigurasi	Keterangan
<i>Batch</i>	4
<i>Width</i>	416
<i>Height</i>	416
<i>Epoch</i>	40
<i>Weights</i>	yolov5l.pt
<i>Classes</i>	5

4.1. Training

Pada proses *training* ini penulis menggunakan Anaconda dengan membuat *environment* baru dengan pengaturan Python versi 3.9.18. Untuk melakukan proses *training* di Anaconda, Ultralytics YOLOv5 sangat diperlukan. Ultralytics YOLOv5 bisa didapatkan melalui *website* <https://github.com/ultralytics/yolov5>.

Untuk mengimplementasikan Ultralytics melalui terminal dari *environment* yang sudah dibuat, lokasikan dulu terminal untuk *compile* di dalam folder YOLOv5 dengan cara ketik “cd (lokasi *file*)” contohnya “cd D:\PROJECT TA\yolov5-master”, kemudian untuk melakukan *install* Ultralytics menggunakan *code* “pip install ultralytics”. Setelah selesai ketiklah *code* untuk melakukan *training* data, sebagai contoh: “python train.py --img 416 --data Vehicle.yaml --batch 4 --epoch 40 --weights yolov5n.pt”

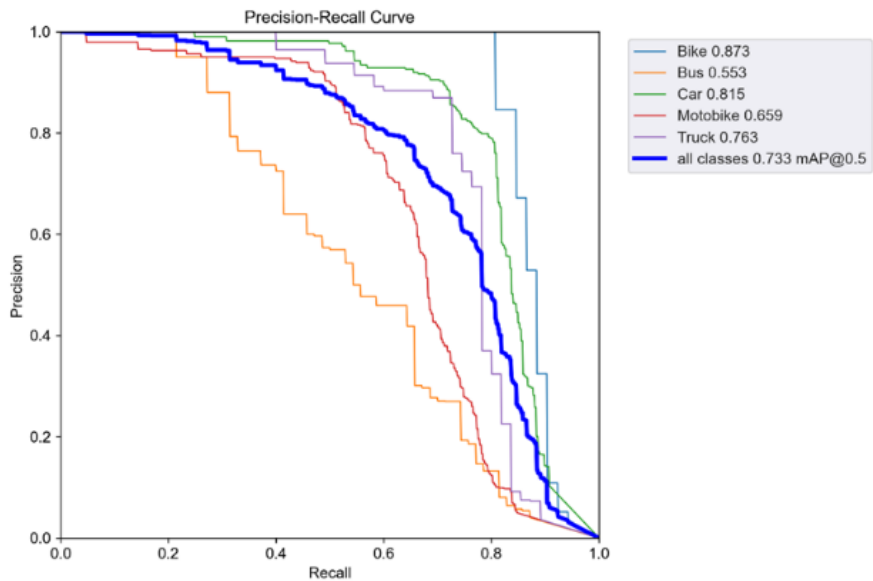
Berkaitan dengan *file* train.py, ini merupakan skrip Python yang penting untuk menginisiasi dan mengeksekusi proses *training* model deteksi objek menggunakan YOLOv5. Dalam konteks ini, skrip tersebut memulai dengan inisialisasi parameter *training* kritis seperti *learning rate*, jumlah *epoch*, dan *batch size*. Selanjutnya, langkah-langkah ini diikuti dengan memuat *dataset* yang telah diannotasi dan membaginya menjadi set *training* dan set validasi.

	Y olov5n			Y olov5s			Y olov5m			Y olov5l		
Class	P	R	mAP P@5	P	R	mAP 50	P	R	mAP 50	P	R	mAP50
Bike	0,667	0,596	0,689	0,714	0,788	0,768	0,843	0,865	0,890	0,807	0,846	0,873
Bus	0,303	0,129	0,139	0,369	0,400	0,373	0,680	0,457	0,551	0,727	0,371	0,553
Car	0,642	0,673	0,670	0,808	0,648	0,732	0,821	0,751	0,820	0,886	0,720	0,815
Motobike	0,537	0,464	0,450	0,523	0,632	0,533	0,809	0,527	0,649	0,829	0,536	0,659
Truck	0,589	0,391	0,393	0,707	0,436	0,509	0,789	0,691	0,723	0,855	0,727	0,763
All	0,548	0,451	0,468	0,646	0,561	0,588	0,788	0,658	0,726	0,821	0,640	0,733

Tabel 7. Training Performance dari YOLOv5 Series

Dalam tahap *training*, penulis mengamati mAP@0.5 untuk mengevaluasi seberapa baik detektor berperforma pada set *training*; nilai yang lebih tinggi menandakan kinerja yang lebih unggul. *File dataset* dengan format Yet Another Markup Language (YAML) mungkin menjadi salah satu aspek terpenting dalam *training* YOLOv5. File ini menyajikan nama-nama kelas dengan jalur menuju data yang digunakan untuk *training* dan validasi. Agar skrip *train* dapat mengenali jalur gambar, jalur label, dan nama-nama kelas, diperlukan penyediaan jalur *file* ini sebagai argumen saat menjalankan skrip *train*. Semua informasi ini sudah tersedia dalam *dataset*. Tabel 7 memberikan gambaran proses *training* untuk seri YOLOv5. YOLOv5n, YOLOv5s, YOLOv5m dan YOLOv5l mencapai rata-rata mAP yang berbeda, yakni 47% untuk YOLOv5n, 59% untuk YOLOv5s, 72% untuk YOLOv5m, diikuti oleh YOLOv5l dengan rata-rata mAP sebesar 73%. Berdasarkan hasil *training*, dapat disimpulkan bahwa YOLOv5l merupakan metode yang paling stabil dibandingkan dengan metode lainnya, dengan tingkat presisi sebesar 82%, *recall* 64%, dan mAP 73%. Selain itu, kelas *Bike* dan kelas *Car* mencapai tingkat akurasi tertinggi untuk semua model, dengan mAP berkisar antara 81% hingga 87%.

Selain perbandingan, *training* juga menghasilkan kurva presisi–*recall*, yang otomatis disimpan untuk setiap *training* di dalam exp. Gambar 8 menggambarkan presisi dan *recall* untuk YOLOv5l. Metrik kinerja digunakan untuk mengevaluasi *dataset Vehicle* dalam eksperimen model YOLOv5 mencakup presisi, *recall*, skor akurasi, dan F1.



Gambar 8. Precision dan Recall YOLOv5l

4.2. Validasi

Validasi data merupakan tahap penting dalam proses pembangunan model deteksi objek. Pada fase ini, model yang telah dilatih dievaluasi menggunakan *dataset* validasi untuk mengukur kinerjanya pada data yang tidak pernah dilihat sebelumnya. Metode umum untuk melakukan validasi pada model deteksi objek, seperti YOLOv5, melibatkan pengukuran beberapa metrik performa, termasuk presisi, *recall*, akurasi, dan *mean Average Precision* (mAP). Hasil validasi adanya pemahaman yang lebih baik kepada sebaik apa model dapat mengenali objek dalam berbagai situasi dan kondisi.

Proses validasi ini melibatkan penggunaan model yang telah dilatih guna prediksi kepada *dataset* validasi. Hasil prediksi ini kemudian dibandingkan dengan anotasi sebenarnya dari *dataset* untuk menghitung metrik performa seperti presisi, *recall*, akurasi, dan mAP.

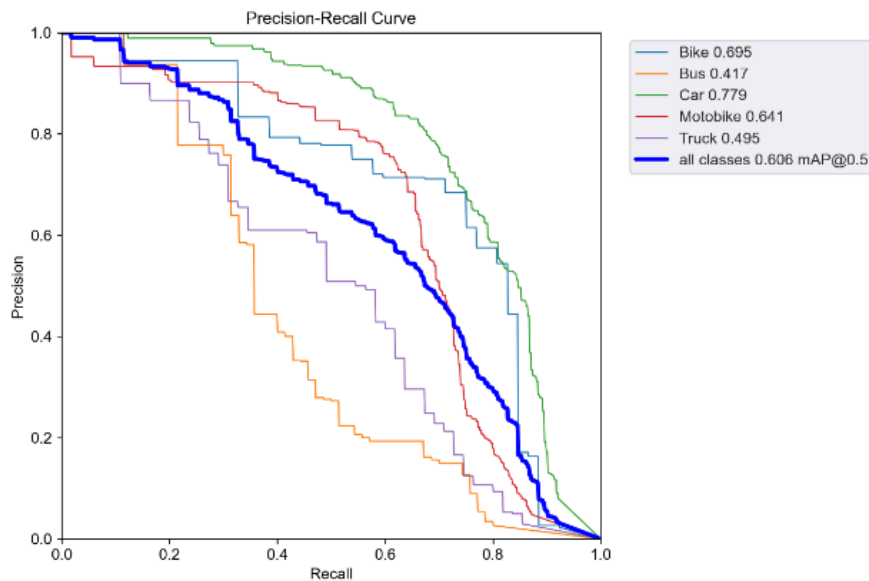
Setelah proses validasi selesai, hasilnya dapat ditinjau melalui beberapa metrik dan tampilan visual yang memberikan wawasan tentang kinerja model. Hasil tersebut disimpan dalam folder *exp* yang berada di folder *val* dalam *runs*. Berikut ini merupakan tabel yang berisi perbandingan data saat penulis melakukan validasi dengan versi YOLOv5 yang berbeda-beda, mulai dari YOLOv5n, YOLOv5s, YOLOv5m, dan YOLOv5l.

	Y olov5n			Y olov5s			Y olov5m			Y olov5l		
Class	P	R	mAP@50	P	R	mAP@50	P	R	mAP@50	P	R	mAP@50
Bike	0,535	0,481	0,489	0,658	0,558	0,768	0,615	0,692	0,652	0,708	0,712	0,695
Bus	0,207	0,057	0,071	0,339	0,214	0,373	0,578	0,329	0,412	0,480	0,357	0,417
Car	0,508	0,622	0,602	0,807	0,588	0,732	0,684	0,726	0,758	0,765	0,702	0,779
Motobike	0,432	0,470	0,414	0,604	0,502	0,533	0,633	0,583	0,603	0,752	0,609	0,641
Truck	0,318	0,109	0,207	0,610	0,127	0,509	0,453	0,382	0,318	0,596	0,364	0,495
All	0,400	0,348	0,357	0,603	0,398	0,588	0,593	0,542	0,549	0,661	0,548	0,606

Tabel 9. Validating Performance dari YOLOv5 Series

Dalam tahap validasi, penulis mengamati mAP@0.5 untuk mengevaluasi seberapa baik detektor berperforma pada set validasi; nilai yang lebih tinggi menandakan kinerja yang lebih unggul. *File dataset* dengan format Yet Another Markup Language (YAML) mungkin menjadi salah satu aspek terpenting dalam *training* YOLOv5. *File* ini menyajikan nama-nama kelas dengan jalur menuju data yang digunakan untuk *training* dan validasi. Agar skrip *val* dapat mengenali jalur gambar, jalur label, dan nama-nama kelas, diperlukan penyediaan jalur *file* ini sebagai argumen saat menjalankan skrip *val*. Semua informasi ini sudah tersedia dalam *dataset*. Tabel 9 memberikan gambaran proses validasi untuk seri YOLOv5. YOLOv5n, YOLOv5s, YOLOv5m dan YOLOv5l mencapai rata-rata mAP yang berbeda, yakni 36% untuk YOLOv5n, 44% untuk YOLOv5s, 55% untuk YOLOv5m, diikuti oleh YOLOv5l dengan rata-rata mAP sebesar 60%. Berdasarkan hasil *training*, dapat disimpulkan bahwa YOLOv5l merupakan metode yang paling stabil dibandingkan dengan metode lainnya, dengan tingkat presisi sebesar 66%, *recall* 55%, dan mAP 60%. Selain itu, kelas *Bike* dan kelas *Car* mencapai tingkat akurasi tertinggi untuk semua model, dengan mAP berkisar antara 64% hingga 78%.

Selain perbandingan, *training* juga menghasilkan kurva *presisi-recall*, yang otomatis disimpan untuk setiap *training* di dalam *exp*. Gambar 10 menggambarkan presisi dan *recall* untuk YOLOv5l. Metrik kinerja digunakan untuk mengevaluasi *dataset Vehicle* dalam eksperimen model YOLOv5 mencakup presisi, *recall*, skor akurasi, dan F1.

Gambar 10. *Precision and Recall YOLOv5l*

4.3. Pengujian

Pada tahap pengujian deteksi, model deteksi objek dievaluasi menggunakan *weight* yang telah di-*train* sebelumnya pada lima kelas tertentu. Tujuan dari pengujian ini adalah untuk menilai tingkat akurasi model dalam mendeteksi objek pada kelas yang telah dibuat sebelumnya. Proses pengujian menggunakan program yang sudah dikembangkan sebelumnya serta memanfaatkan model *weight*. Dengan memanfaatkan *weight* yang telah disesuaikan dengan lima kelas yang ditentukan, pengguna dapat mengevaluasi kemampuan model dalam mengenali dan mendeteksi objek pada kelas-kelas yang telah diidentifikasi.

Dengan menjalankan perintah ini, model akan melakukan deteksi pada gambar "img30.jpg" dengan menggunakan *weight* terbaik yang telah di-*train* sebelumnya. Hasil deteksi, termasuk kotak pembatas dan label objek, akan disimpan dalam *file teks* untuk dianalisis lebih lanjut atau digunakan dalam tahap berikutnya, hasil pengujian yakni pada Tabel 3.

Tabel 3. Hasil Pengujian

<i>File</i>	<i>Bike</i>	<i>Bus</i>	<i>Car</i>	<i>Motobike</i>	<i>Truck</i>	Waktu Proses
img1	87%	-	-	-	-	2,13 ms
img2	-	-	-	72%	-	3,37 ms
img3	-	-	-	-	91%	3,22 ms
img4	-	91%	-	-	-	3,17 ms
img5	-	-	89%	-	-	4,45 ms
img6	91%	-	91%	88%	-	2,03 ms

Pada *detecting file* pertama yaitu *img1*, didapatkan nilai *confidence* pada kelas *Bike* sebesar 87%. Pada *img2* nilai *confidence* pada kelas *Motobike* sebesar 72%, sedangkan untuk *img6* didapatkan nilai untuk tiga kelas yaitu kelas *MotoBike* sebesar 88% lalu *Bike* dan *Car* sebesar 91%. Contoh hasil deteksi yakni pada Gambar 11 serta Gambar 12.



Gambar 11. Hasil Deteksi pada img4



Gambar 12. Hasil Deteksi pada img6

5. Kesimpulan dan Saran

Pada observasi yang dilakukan, disimpulkan bahwa algoritma *YOLO* (You Only Look Once) mampu melakukan deteksi kendaraan pada kelas *Bike*, *Bus*, *Car*, *Motobike*, dan *Truck* dengan kinerja yang baik. Hasil deteksi pada foto menunjukkan variasi nilai *confidence* pada setiap *frame*, yang disebabkan oleh perbedaan posisi objek dalam berbagai kondisi. Analisis data dari proses pelatihan menunjukkan bahwa jumlah *True Positive* jauh lebih besar dibanding pada jumlah *False Positive*, menandakan bahwa sistem bisa mendeteksi objek dengan efektif. Selain itu, nilai *Average Precision* (AP) cukup tinggi untuk setiap kelas, seperti 90% untuk mobil, 80% untuk sepeda motor, 89% untuk sepeda, 91% untuk truk, dan 91% untuk bus. Secara keseluruhan, nilai *mean Average Precision* (mAP) mencapai 87,25%, dengan waktu pemrosesan sekitar 0,3 detik per *frame*. Pentingnya kualitas foto juga terbukti mempengaruhi tingkat keberhasilan deteksi, di mana semakin tinggi kualitas foto, hasil klasifikasi serta *bounding box* makin optimal.

Berdasarkan hasil penelitian, penulis memiliki beberapa saran pada masa mendatang. Pertama, disarankan untuk meningkatkan jumlah dan kualitas data pada tiap kelas yang dipakai dalam proses pelatihan guna memaksimalkan tingkat akurasi pada model *weights*. Selain itu, direkomendasikan untuk mengembangkan tampilan antarmuka yang intuitif dan *user-friendly* agar memudahkan pengguna dalam menjalankan operasi sistem. Perlu juga dilakukan penelitian lebih lanjut untuk mengeksplorasi dan mengatasi tantangan dalam mendeteksi objek pada kondisi lalu lintas yang padat. Hasil penelitian ini, juga dapat digunakan atau diimplementasikan di kamera-kamera CCTV yang ada di jalan raya, parkir dan sebagainya untuk membantu mengidentifikasi atau menghitung jumlah kendaraan dengan mudah.

Referensi

- [1] R. Kurniawan, A. T. Martadinata, and S. D. Cahyo, "Klasifikasi Tingkat Kematangan Buah Sawit Berbasis Deep Learning dengan Menggunakan Arsitektur YOLOv5," *Journal of Information System Research (JOSH)*, vol. 5, no. 1, pp. 302-309, 2023.
- [2] H. Dawami, E. Rachmawati, and M. D. Sulistiyo, "Deteksi Penggunaan Masker Wajah Menggunakan YOLOv5," *eProceedings of Engineering*, vol. 10, no. 2, 2023.
- [3] N. A. K. D. Pasongko, A. Khairunnisa and S. Aras, "Deteksi Penggunaan Safety Helmet Menggunakan YOLOv5," *JIEET (Journal of Information Engineering and Educational Technology)*, vol. 7, no. 2, pp. 74-77, 2023.
- [4] C. Dewi, R. C. Chen, Y. C. Zhuang, H. J. Christanto, "YOLOv5 series algorithm for road marking sign identification," *Big Data and Cognitive Computing*, vol. 6, no. 4, pp. 149, 2022.
- [5] Z. Wang, L. Jin, S. Wang, H. Xu, "Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system," *Postharvest Biology and Technology*, vol. 185, pp. 111808, 2022.
- [6] D. I. Mulyana and M. A. Rofik, "Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOV5," *Jurnal Pendidikan Tambusai*, vol. 6, no. 3, pp. 13971-13982, 2022.
- [7] J. Yao, J. Qi, J. Zhang, H. Shao, J. Yang, and X. Li, "A real-time detection algorithm for Kiwifruit defects based on YOLOv5," *Electronics*, vol. 10, no. 14, pp. 1711, 2021.
- [8] J. S. W. Hutaeruk, T. Matulatan, and N. Hayaty, "Deteksi kendaraan secara real time menggunakan metode YOLO berbasis android," *Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan*, vol. 9, no. 1, pp. 8-14, 2020.
- [9] M. Ju, H. Luo, and Z. Wang, "An improved YOLO V3 for small vehicles detection in aerial images," in *Proceedings of the 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence*, Dec. 2020, pp. 1-5.
- [10] F. A. Khan, N. Nagori, and A. Naik, "Helmet and Number Plate Detection of Motorcyclists Using Deep Learning and Advanced Machine Vision Techniques," in *Proceedings of the 2nd International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2020, pp. 714-717.
- [11] C. Ding, et al., "REQ-YOLO: A Resource-Aware, Efficient Quantization Framework for Object Detection on FPGAs," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2019, pp. 33-42.