

Evaluasi Efisiensi dan Performa YOLOv8 Small-Scale Variants dalam Pendeteksian Jalan Berlubang

Adam Fitriyono¹, Mohammad Tsaqif Akmal Al Hamam², Bagus Wahyu Saputra³, Muhammad Naufal^{4,*}, Ricardus Anggi Pramunendar⁵

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia
Email: ¹11202315340@mhs.dinus.ac.id, ²11202314899@mhs.dinus.ac.id, ³11202314157@mhs.dinus.ac.id,
^{4,*}m.naufal@dsn.dinus.ac.id, ⁵ricardus.anggi@dsn.dinus.ac.id
Email Penulis Korespondensi: m.naufal@dsn.dinus.ac.id

Abstrak—Kerusakan jalan berlubang merupakan salah satu permasalahan utama dalam pemeliharaan infrastruktur jalan di Indonesia. Proses inspeksi manual yang umum digunakan masih membutuhkan waktu, biaya, dan tenaga yang besar sehingga kurang efisien untuk pemantauan secara berkelanjutan. Penelitian ini mengembangkan sistem deteksi jalan berlubang berbasis deep learning menggunakan dua varian lightweight model YOLOv8, yaitu YOLOv8s dan YOLOv8n, untuk membandingkan performa deteksi dan efisiensi komputasi keduanya. Dataset yang digunakan terdiri dari 1.330 citra jalan berlubang dengan variasi ukuran, bentuk, pencahayaan, dan kondisi permukaan jalan. Evaluasi dilakukan menggunakan metrik mAP50, mAP50-95, recall, F1-score, inference time, dan frame per second (FPS). Hasil penelitian menunjukkan bahwa YOLOv8s memberikan performa deteksi yang lebih tinggi dengan nilai mAP50 sebesar 0.8060, mAP50-95 sebesar 0.5595, recall sebesar 0.7023, dan F1-score sebesar 0.7584. Namun, model ini memiliki waktu inferensi lebih lama, yaitu 13.44 ms per frame. Sementara itu, YOLOv8n menunjukkan kecepatan inferensi yang lebih baik dengan waktu 11.96 ms per frame dan FPS sebesar 83.63, meskipun memiliki akurasi yang sedikit lebih rendah dengan mAP50 sebesar 0.7926. Hasil penelitian menunjukkan adanya trade-off antara akurasi deteksi dan efisiensi komputasi pada kedua model. YOLOv8n lebih sesuai untuk implementasi real-time pada perangkat dengan keterbatasan sumber daya, sedangkan YOLOv8s lebih cocok untuk aplikasi yang memprioritaskan akurasi deteksi. Penelitian ini diharapkan dapat mendukung pengembangan sistem pemantauan kerusakan jalan secara otomatis dan real-time.

Kata Kunci: Deteksi Jalan Berlubang; Performa Deteksi; YOLOv8; Objek Deteksi; FPS

Abstract—Pothole damage to roads is one of the main problems in maintaining road infrastructure in Indonesia. The manual inspection process that is commonly used still requires a large amount of time, money and energy, making it less efficient for continuous monitoring. This research develops a deep learning-based pothole detection system using two lightweight variants of the YOLOv8 model, namely YOLOv8s and YOLOv8n, to compare the detection performance and computational efficiency of the two. The dataset used consists of 1,330 images of potholes with variations in size, shape, lighting and road surface conditions. Evaluation was carried out using the metrics mAP50, mAP50-95, recall, F1-score, inference time, and frames per second (FPS). The research results show that YOLOv8s provides higher detection performance with a mAP50 value of 0.8060, mAP50-95 of 0.5595, recall of 0.7023, and F1-score of 0.7584. However, this model has a longer inference time, namely 13.44 ms per frame. Meanwhile, YOLOv8n shows better inference speed with a time of 11.96 ms per frame and FPS of 83.63, although it has slightly lower accuracy with mAP50 of 0.7926. The research results show that there is a trade-off between detection accuracy and computational efficiency in both models. YOLOv8n is better suited for real-time implementation on resource-constrained devices, while YOLOv8s is better suited for applications that prioritize detection accuracy. It is hoped that this research can support the development of an automatic and real-time road damage monitoring system.

Keywords: Pothole Detection; Detection Performance; YOLOv8; Object Detection FPS

1. PENDAHULUAN

Kerusakan jalan berlubang menjadi masalah serius bagi keselamatan pengguna jalan di Indonesia. Penelitian menunjukkan bahwa jalan berlubang merupakan salah satu faktor penyebab kecelakaan lalu lintas yang signifikan [1], terutama bagi pengendara sepeda motor yang rentan mengalami kecelakaan akibat kondisi jalan yang buruk. Kerusakan jalan seperti lubang tidak hanya mengganggu kelancaran lalu lintas, tetapi juga menimbulkan risiko kecelakaan dan kerusakan pada kendaraan [2]. Saat ini, proses deteksi dan perbaikan kerusakan jalan masih mengandalkan inspeksi manual, yang tidak hanya memakan waktu dan biaya besar, tetapi juga kurang efisien dan berisiko terlewatkan dalam deteksi. Oleh karena itu, solusi yang lebih efisien dan akurat diperlukan untuk meningkatkan pemeliharaan jalan dan mempercepat proses deteksi kerusakan. Proses pemeliharaan jalan yang lambat dan biaya tinggi yang terkait dengan inspeksi manual menjadi isu utama dalam penanganan kerusakan jalan [2]. Selain itu, kurangnya sistem pemantauan yang efisien juga menyebabkan keterlambatan dalam penanganan kerusakan yang dapat semakin parah. Isu lainnya adalah ketergantungan pada kualitas gambar yang diambil saat inspeksi, yang sering kali terpengaruh oleh kondisi cuaca dan keterbatasan teknologi yang digunakan [9]. Teknologi berbasis visi komputer, seperti deteksi objek dengan algoritma *deep learning*, menawarkan potensi besar dalam mengatasi masalah ini dengan mendeteksi kerusakan jalan secara otomatis dan real-time menggunakan citra visual dari CCTV atau perangkat monitoring lainnya [3].

Berdasarkan permasalahan tersebut, penting untuk mengembangkan sistem deteksi otomatis yang dapat melakukan identifikasi jalan berlubang secara cepat dan akurat. Salah satu teknologi yang menjanjikan adalah penggunaan model deteksi objek berbasis YOLOv8 (You Only Look Once), yang telah dikenal efektif dalam tugas deteksi objek dengan akurasi dan kecepatan tinggi [4],[12],[5]. YOLOv8 memiliki berbagai peningkatan dalam hal kinerja dibandingkan dengan versi sebelumnya [6] dengan versi kecil (YOLOv8n) yang lebih ringan dan lebih cepat, serta versi

lebih besar (YOLOv8s) yang lebih unggul dalam akurasi deteksi [7]. Penelitian terkait deteksi jalan berlubang menggunakan teknologi visi komputer telah dilakukan sebelumnya. Salah satunya adalah penelitian yang menggunakan Convolutional Neural Networks (CNN) untuk deteksi jalan berlubang, dengan akurasi 88%. Namun, metode CNN ini masih terbatas dalam hal waktu pemrosesan dan ketergantungan pada kualitas citra. Penelitian lain mengembangkan sistem deteksi jalan berlubang menggunakan YOLOv8, hasil penelitian menunjukkan YOLOv8 efektif dalam mendeteksi kerusakan jalan dengan akurasi yang tinggi, terutama dalam pembagian data 90%:10% yang menghasilkan kepercayaan 97% dan mAP 93.2% pada epoch ke-100 [21]. Tujuan utama dari penelitian tersebut adalah untuk mengevaluasi performa kedua model dalam mendeteksi jalan berlubang secara otomatis dan untuk memberikan rekomendasi model yang lebih efisien dalam pemantauan jalan berbasis sistem CCTV atau perangkat monitoring lainnya [8].

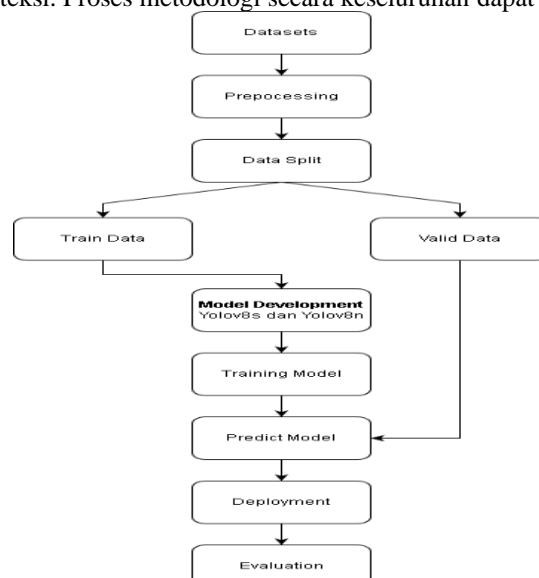
Penelitian ini bertujuan untuk mengembangkan sistem deteksi otomatis jalan berlubang menggunakan dua model YOLOv8, yakni YOLOv8s dan YOLOv8n, dan membandingkan kinerja keduanya dalam hal akurasi dan kecepatan deteksi [7]. Meskipun telah tersedia versi YOLO yang lebih baru seperti YOLOv10 dan YOLOv11, penelitian ini memilih YOLOv8 karena arsitektur tersebut masih banyak digunakan sebagai baseline pada berbagai penelitian object detection akibat stabilitas, dokumentasi yang matang, serta implementasi yang ringan dan efisien untuk perangkat dengan keterbatasan komputasi. Selain itu, YOLOv8 menyediakan varian lightweight seperti YOLOv8n dan YOLOv8s yang dirancang untuk kebutuhan real-time detection dengan keseimbangan antara akurasi dan efisiensi inferensi. Pemilihan YOLOv8 pada penelitian ini bertujuan untuk mengevaluasi trade-off performa dan efisiensi komputasi pada skenario deteksi pothole, sehingga fokus penelitian tidak terletak pada penggunaan arsitektur terbaru, melainkan pada analisis karakteristik performa model lightweight yang lebih realistis untuk implementasi lapangan.

Dengan mengintegrasikan teknologi ini, diharapkan pemeliharaan jalan dapat dilakukan dengan lebih efisien, mengurangi ketergantungan pada inspeksi manual, serta meningkatkan keselamatan dan kenyamanan pengendara [2]. Penelitian ini membedakan diri dengan menguji dua model YOLOv8, yaitu YOLOv8s dan YOLOv8n. YOLOv8s memiliki keunggulan dalam hal akurasi deteksi yang lebih tinggi, sedangkan YOLOv8n menawarkan kecepatan deteksi yang lebih tinggi dengan ukuran model yang lebih kecil dan lebih efisien dalam penggunaan sumber daya [4]. Dengan membandingkan kedua model ini [16], penelitian ini diharapkan dapat memberikan kontribusi dalam menentukan model yang lebih tepat untuk digunakan dalam aplikasi pemantauan jalan berbasis perangkat monitoring dengan keterbatasan sumber daya.

Kontribusi utama dari penelitian ini adalah memberikan perbandingan langsung antara dua model deteksi objek berbasis YOLOv8 yang paling terbaru dalam konteks deteksi jalan berlubang, yang dapat diintegrasikan dalam sistem pemantauan real-time untuk perbaikan jalan [22]. Penelitian ini juga diharapkan dapat memberikan kontribusi signifikan dalam meningkatkan efisiensi pemeliharaan infrastruktur jalan dan keselamatan transportasi di Indonesia [12]. Adapun keterbatasan penelitian ini terletak pada penggunaan dataset terbatas yang hanya mencakup gambar jalan berlubang, sehingga tidak dapat mencakup deteksi kerusakan jalan lainnya seperti retak atau deformasi struktural. Selain itu, penelitian ini terbatas pada penggunaan perangkat keras dengan spesifikasi tertentu, yang mungkin mempengaruhi hasil deteksi di perangkat dengan spesifikasi lebih rendah.

2. METODOLOGI PENELITIAN

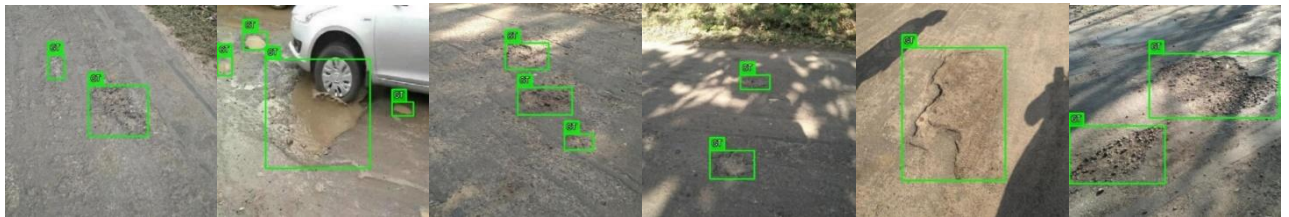
Penelitian ini bertujuan untuk membandingkan kinerja dua model deteksi objek berbasis YOLOv8, yaitu YOLOv8s (small) dan YOLOv8n (nano), dalam mendeteksi jalan berlubang secara otomatis. Untuk mencapai tujuan tersebut, penelitian ini melalui beberapa tahapan yang mencakup pemilihan dataset, pra-pemrosesan data, pelatihan model, serta evaluasi performa deteksi. Proses metodologi secara keseluruhan dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

2.1 Pengumpulan Data

Pada penelitian ini, dataset yang digunakan untuk pelatihan dan evaluasi model deteksi jalan berlubang diperoleh dari platform Roboflow, dataset yang digunakan adalah "Pothole Detection Dataset" yang dikurasi oleh pengguna RUCHI dan dipublikasikan di Roboflow Universe [13], yang menyediakan dataset gambar dengan anotasi objek (bounding box). Dataset yang digunakan terdiri dari 1.330 gambar jalan berlubang yang telah terlabel dengan satu kelas objek yaitu 'pothole'. Dataset ini mencakup gambar dengan kondisi jalan berlubang yang beragam, termasuk variasi dalam ukuran, bentuk, dan kedalaman kerusakan. Selain itu, gambar-gambar ini diambil dari berbagai lokasi yang mewakili kondisi jalan berlubang di Indonesia, baik di wilayah perkotaan maupun pedesaan, dengan variasi kondisi cuaca dan pencahayaan, sehingga memberikan representasi yang lebih umum terhadap kerusakan jalan di Indonesia. Hasil dari pengumpulan gambar jalan berlubang dengan bounding box ditunjukkan pada Gambar 2.



Gambar 2. Sampel Dataset

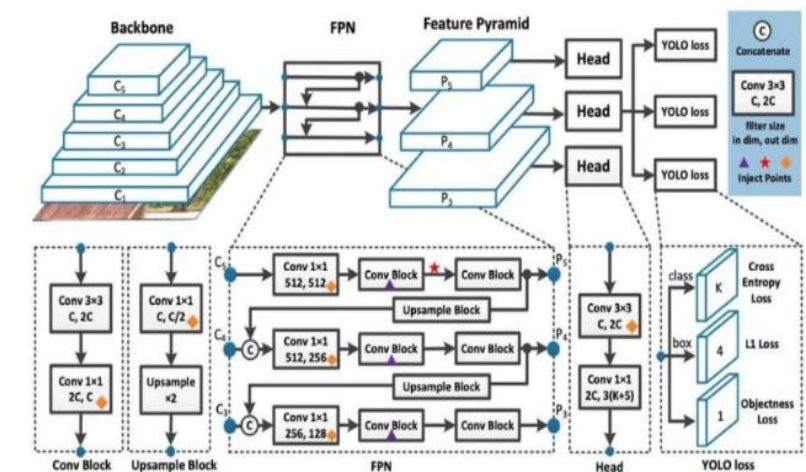
2.2 Pra-Pemrosesan Data

Pra-pemrosesan data merupakan tahapan krusial untuk mempersiapkan dataset sebelum proses pelatihan model dilakukan. Tahapan pertama dalam pra-pemrosesan adalah melakukan pembagian dataset menjadi training set dan validation set. Pada penelitian ini, dataset dibagi dengan perbandingan 80:20, di mana 80% digunakan untuk melatih model dan 20% untuk memvalidasi performa model. Perincian pembagian data ditunjukkan pada Tabel 1.

Tabel 1. Pembagian Data

Kategori	Jumlah Gambar	Presentase
Training Set	1064	80%
Validation Set	266	20%
Total	1330	100%

Tahap selanjutnya melibatkan penyesuaian ukuran semua gambar menjadi resolusi standar 416×416 piksel. Proses resize ini diperlukan untuk memastikan konsistensi input ke model dan mempercepat proses komputasi selama pelatihan. Setelah proses resize, dataset siap untuk digunakan dalam pelatihan kedua model YOLOv8s dan YOLOv8n. Pada penelitian ini, augmentasi data diterapkan secara otomatis selama proses training menggunakan fitur augmentasi bawaan YOLOv8. Teknik augmentasi yang diterapkan mencakup horizontal flip, vertical flip, rotasi acak, brightness adjustment, contrast adjustment, saturation adjustment, dan mosaic augmentation. Penerapan augmentasi data bertujuan untuk memperkaya variasi dataset dan meningkatkan kemampuan model dalam menggeneralisasi terhadap berbagai kondisi visual jalan berlubang di dunia nyata, sekaligus mengurangi risiko overfitting pada data training [14], [15].



Gambar 3. Arsitektur YOLOv8 [16]

YOLOv8 (You Only Look Once version 8) adalah algoritma deteksi objek generasi terbaru yang dikembangkan oleh Ultralytics sebagai penyempurnaan dari versi sebelumnya [14],[12]. YOLOv8 termasuk dalam kategori one-stage detector yang mampu melakukan prediksi lokasi dan kelas objek secara bersamaan dalam satu pass forward, berbeda dengan two-stage detector yang memerlukan dua tahap deteksi [14]. Cara kerja YOLOv8 adalah dengan membagi gambar input menjadi grid cells dan memprediksi bounding box serta confidence score untuk setiap cell secara paralel, menghasilkan deteksi yang cepat dan efisien. Keunggulan utama YOLOv8 adalah kombinasi kecepatan tinggi dengan akurasi yang baik, arsitektur yang lebih efisien, kemudahan dalam implementasi, dan dukungan untuk berbagai ukuran model yang dapat disesuaikan dengan kebutuhan aplikasi. YOLOv8 menawarkan beberapa varian model dengan ukuran dan kompleksitas berbeda untuk fleksibilitas deployment [3],[14].

Tabel 2. Spesifikasi YOLOv8s dan YOLOv8n

Spesifikasi	YOLOv8s	YOLOv8n
Versi	Small	Nano
Layer	129	129
Parameter	11,135,987	3,011,043
Gradients	11,135,971	3,011,027
(GFLOPs)	28.6	8.2

Penelitian ini menggunakan dua varian YOLOv8, yaitu YOLOv8s (Small) dan YOLOv8n (Nano) yang dimuat pada Tabel 2, yang dipilih berdasarkan trade-off antara akurasi dan efisiensi [14]. YOLOv8s adalah varian berukuran sedang yang menawarkan keseimbangan antara akurasi deteksi dan kecepatan pemrosesan. Model YOLOv8s memiliki arsitektur dengan 129 layers, 11,135,987 parameters, 11,135,971 gradients, dan computational complexity sebesar 28.6 GFLOPs. Varian ini cocok untuk aplikasi yang memerlukan akurasi tinggi dengan resource komputasi yang masih reasonable. Sebaliknya, YOLOv8n adalah varian berukuran kecil (Nano) yang dioptimalkan untuk kecepatan dan efisiensi resource. Model YOLOv8n memiliki arsitektur dengan 129 layers (sama dengan YOLOv8s), tetapi dengan parameter yang jauh lebih sedikit yaitu 3,011,043 parameters, 3,011,027 gradients, dan computational complexity hanya 8.2 GFLOPs. Pemilihan kedua varian ini memungkinkan penelitian untuk melakukan analisis trade-off yang komprehensif antara performa akurasi deteksi dan efisiensi komputasi pada tugas deteksi jalan berlubang [9], [10].

2.3 Pelatihan Model

Tahapan pelatihan model dilakukan dengan dataset yang telah dipersiapkan melalui platform Roboflow. Platform ini menyediakan fitur untuk mengeksplor dataset dalam format yang kompatibel dengan YOLOv8, memuat informasi path training set, validation set, dan nama kelas objek, yang memungkinkan dataset langsung dapat digunakan dalam proses pelatihan di lingkungan Google Colab. Konfigurasi pelatihan model dirancang untuk memastikan performa yang optimal dan perbandingan yang fair antara kedua varian model. Tabel 3 menampilkan parameter-parameter training yang digunakan dalam penelitian ini.

Tabel 3. Parameter Pelatihan Model YOLOv8s Dan YOLOv8n

Parameter	Nilai
Epochs	50
Image Size	416
Batch Size	32
Learning Rate (lr0)	0.01
Learning Rate (lrf)	0.1
Warmup Epochs	3
Momentum	0.937
Weight Decay	0.0005
Augmentation	True
Workers	4

Berdasarkan tabel di atas, kedua model YOLOv8s dan YOLOv8n dilatih menggunakan parameter yang identik untuk memastikan perbandingan yang objektif. Pelatihan dilakukan selama 50 epoch dengan batch size sebesar 32, yang merupakan keseimbangan antara penggunaan memori GPU dan stabilitas gradient descent. Ukuran gambar input distandarkan menjadi 416×416 piksel untuk menjaga konsistensi input ke kedua model. Learning rate diatur dengan nilai awal 0.01 yang akan secara bertahap menurun menjadi 0.1 dari nilai awal, dengan warmup epochs sebesar 3 untuk stabilisasi pembelajaran pada tahap awal training. Momentum dan weight decay diatur sesuai dengan nilai yang telah terbukti efektif untuk pelatihan model YOLOv8 [10]. Augmentasi data diaktifkan untuk meningkatkan robustness model [19]. Pelatihan dilakukan di lingkungan Google Colab dengan GPU Tesla T4 untuk akselerasi komputasi. Kedua model diinisialisasi dengan pre-trained weights dari dataset COCO untuk memanfaatkan transfer learning, yang memungkinkan model untuk beradaptasi lebih cepat terhadap tugas spesifik deteksi jalan berlubang.

2.4 Deployment

Tahap deployment merupakan fase implementasi model hasil pelatihan ke dalam lingkungan inferensi untuk memvalidasi kemampuannya beroperasi pada data uji maupun skenario dunia nyata. Proses ini diawali dengan inialisasi sistem yang memuat bobot model terbaik (best.pt), yang dipilih berdasarkan checkpoint dengan nilai fitness tertinggi kombinasi dari mAP dan Recall selama fase validasi guna menjamin performa optimal [14]. Agar deteksi berjalan efektif, konfigurasi parameter inferensi diatur dengan menyeimbangkan sensitivitas dan presisi. Confidence threshold ditetapkan pada nilai standar (misalnya 0.25) untuk menyaring prediksi berprobabilitas rendah, sementara algoritma Non-Maximum Suppression (NMS) diterapkan untuk mengeliminasi bounding box yang tumpang tindih (redundant), memastikan sistem hanya menghasilkan satu deteksi paling akurat untuk setiap objek lubang [14].

Dalam alur operasionalnya, data masukan berupa frame video yang di ambil secara realtime melalui pipeline pra-pemrosesan yang konsisten dengan fase pelatihan, meliputi penyesuaian resolusi otomatis (resizing) ke 416x416 piksel dan normalisasi nilai piksel. Model kemudian melakukan eksekusi forward pass untuk memprediksi koordinat kerusakan serta kelas objek, yang selanjutnya divisualisasikan kembali ke citra asli lengkap dengan label kelas dan skor kepercayaannya. Seluruh mekanisme ini diintegrasikan ke dalam antarmuka sistem berbasis web yang memungkinkan akses publik, di mana metrik efisiensi seperti inference time dan Frames Per Second (FPS) turut dipantau untuk mengevaluasi kelayakan komputasi model pada perangkat pengguna.

2.4 Evaluasi Model

Evaluasi model merupakan tahapan penting untuk mengukur performa kedua model YOLOv8s dan YOLOv8n setelah proses pelatihan selesai. Pada tahapan ini, performa model dianalisis menggunakan berbagai metrik evaluasi untuk mengukur kualitas deteksi jalan berlubang. Metrik yang digunakan mencakup confusion matrix untuk memperoleh nilai precision, recall, mAP, akurasi deteksi, serta inference time dan FPS untuk mengukur efisiensi komputasi model [20]. Recall merupakan metrik yang mengukur proporsi objek yang berhasil dideteksi terhadap total objek yang sebenarnya ada, sementara Precision mengukur proporsi prediksi positif yang benar terhadap total prediksi positif yang dilakukan oleh model. Rumus precision dan recall dijelaskan dipersamaan (1) dan persamaan (2).

$$Recall = \frac{TP}{TP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Dalam rumus di atas, TP (True Positive) merepresentasikan total prediksi yang benar, yaitu ketika objek jalan berlubang terdeteksi oleh model. FP (False Positive) merepresentasikan prediksi positif yang salah, yaitu ketika model salah mengidentifikasi area *background* sebagai jalan berlubang. FN (False Negative) merupakan objek jalan berlubang yang sebenarnya ada tetapi tidak terdeteksi oleh model.

$$mAP5 = \frac{1}{N} \sum_{k=0}^{k=n} AP_k \quad (3)$$

$$mAP50 = \frac{1}{N} \sum_{k=0}^{k=n} AP_{50}^k \quad (4)$$

$$mAP50 - 95 = \frac{1}{N} \sum_{k=0}^{k=n} AP_{50-95}^k \quad (5)$$

Mean average precision (mAP) merupakan metrik evaluasi yang mengukur rata-rata *precision* pada berbagai tingkat *recall* untuk mengukur performa deteksi objek secara keseluruhan [21]. di mana N merepresentasikan jumlah total kelas dalam dataset, dan k adalah indeks kelas ke-k. *mAP50* merupakan *mAP* yang dievaluasi pada *IoU (Intersection over union)* threshold sebesar 0.50, sementara *mAP50-95* merupakan *mAP* yang dievaluasi pada berbagai *IoU* threshold mulai dari 0.50 hingga 0.95 dengan interval 0.05. Metrik *mAP50-95* memberikan evaluasi yang lebih komprehensif terhadap kualitas deteksi pada berbagai tingkat *IoU* threshold.

$$L_{box} = \sum_{i=0}^N 1_{obj}^i [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2] \quad (6)$$

$$L_{cls} = \sum_{i=0}^N 1_{obj}^i [ci \log(\hat{ci}) + (1 - ci) \log(1 - ci)] \quad (7)$$

$$L_{dfl} = \sum_{i=0}^N 1_{obj}^i \left[- \sum_j p(y_i) \log(\hat{p}(y_j)) \right] \quad (8)$$

YOLOv8 menggunakan kombinasi tiga komponen *loss function* untuk melatih model. Box Loss, Classification Loss, dan Distribution Focal Loss, ketiga komponen bekerja bersama untuk mengoptimalkan lokalisasi *bounding box*, klasifikasi objek, dan distribusi prediksi. Box Loss (Localization Loss) mengukur akurasi prediksi lokasi dan ukuran *bounding box*. Box loss yang rendah menunjukkan model dapat dengan baik memprediksi lokasi dan ukuran jalan berlubang. Classification Loss (cls_loss) mengukur akurasi klasifikasi kelas objek menggunakan binary cross-entropy. Classification loss yang rendah menunjukkan model memiliki confidence tinggi dalam mengidentifikasi *pothole*. Distribution Focal Loss (DFL) meningkatkan presisi lokalisasi dengan memperlakukan prediksi sebagai distribusi probabilitas. F1-Score merupakan harmonic mean dari *precision* dan *recall* yang memberikan keseimbangan antara kedua metrik tersebut. F1-Score dihitung dengan rumus yang dijelaskan pada persamaan (9).

$$F1 - Score = 2x \frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

Inference time mengukur waktu yang dibutuhkan model untuk melakukan prediksi pada satu gambar *input*, dinyatakan dalam satuan milidetik (ms). *Inference time* yang lebih pendek menunjukkan model yang lebih efisien dalam melakukan deteksi. *Frames per second* (FPS) merupakan metrik yang mengukur jumlah gambar yang dapat diproses oleh model dalam satu detik.

$$FPS = 2x \frac{1000}{Inference\ time\ (ms)} \quad (10)$$

FPS yang lebih tinggi mengindikasikan kemampuan model dalam memproses gambar lebih cepat, yang penting untuk aplikasi *real-time* seperti sistem monitoring jalan berlubang. Model Size merepresentasikan ukuran file model dalam satuan megabyte (MB), yang mencerminkan jumlah parameter dan kompleksitas model. Model yang lebih kecil lebih mudah untuk didistribusikan, disimpan, dan dijalankan pada perangkat dengan keterbatasan memori.

2.5 Analisis Performa

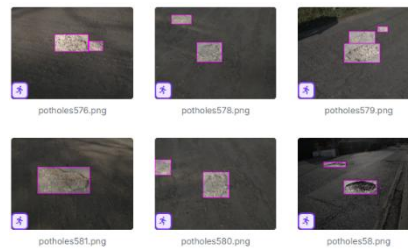
Tahap analisis performa dilakukan untuk membandingkan hasil deteksi jalan berlubang antara model YOLOv8s dan YOLOv8n. Analisis ini mencakup evaluasi akurasi deteksi melalui metrik *precision*, *recall*, *mAP*, dan F1-Score, serta evaluasi efisiensi melalui metrik *inference time*, FPS, dan model size [20], [22]. Tujuannya adalah mengidentifikasi kelebihan dan kekurangan setiap model untuk menentukan model mana yang paling cocok digunakan. Pada aspek akurasi deteksi, *precision* mengukur seberapa akurat prediksi positif model, *recall* mengukur kemampuan model mendeteksi semua jalan berlubang, dan *mAP* mengukur performa deteksi secara keseluruhan [11], [12]. F1-Score memberikan keseimbangan antara *precision* dan *recall*. Pada aspek efisiensi, *inference time* mengukur waktu pemrosesan per gambar, FPS mengukur jumlah gambar yang dapat diproses per detik, dan model size mengukur ukuran file model. Metrik-metrik ini penting untuk mengevaluasi apakah model dapat berjalan dengan cepat dan efisien di lapangan [4]. Analisis antara akurasi dan efisiensi menunjukkan bahwa YOLOv8s diperkirakan akan menghasilkan akurasi lebih tinggi tetapi dengan kecepatan lebih lambat dan ukuran model lebih besar. Sebaliknya, YOLOv8n diperkirakan lebih cepat dan efisien tetapi dengan akurasi yang sedikit lebih rendah. Pemilihan model tergantung pada prioritas aplikasi.

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dari penelitian perbandingan performa YOLOv8s dan YOLOv8n dalam deteksi jalan berlubang. Hasil yang ditampilkan mencakup proses pengumpulan data, hasil pelatihan model, evaluasi akurasi deteksi, evaluasi efisiensi komputasi, serta analisis perbandingan performa kedua model. Setiap hasil akan dianalisis secara detail untuk memberikan pemahaman mendalam tentang karakteristik dan performa masing-masing model dalam tugas deteksi jalan berlubang.

3.1 Hasil Pra-pemrosesan Data

Pada tahap pra-pemrosesan, langkah pertama yang dilakukan adalah anotasi pada citra jalan berlubang untuk mengidentifikasi area yang mengalami kerusakan. Anotasi ini memiliki tujuan untuk mempersiapkan dataset yang akan digunakan dalam proses pelatihan model deteksi berbasis deep learning. Setiap citra jalan berlubang diberi label dengan kotak pembatas (*bounding box*) yang menandakan lokasi lubang pada jalan. Proses anotasi ini tidak hanya bertujuan untuk meningkatkan akurasi deteksi model, tetapi juga untuk memastikan bahwa dataset yang digunakan mencakup berbagai variasi yang diperlukan, yang akan memungkinkan pelatihan model yang lebih efektif. Dengan demikian, model dapat belajar dari beragam contoh dan kondisi yang mungkin ditemui di dunia nyata. Gambar 4 berikut menunjukkan contoh hasil anotasi pada kelas '*pothole*' (jalan berlubang) yang dilakukan menggunakan platform Roboflow.

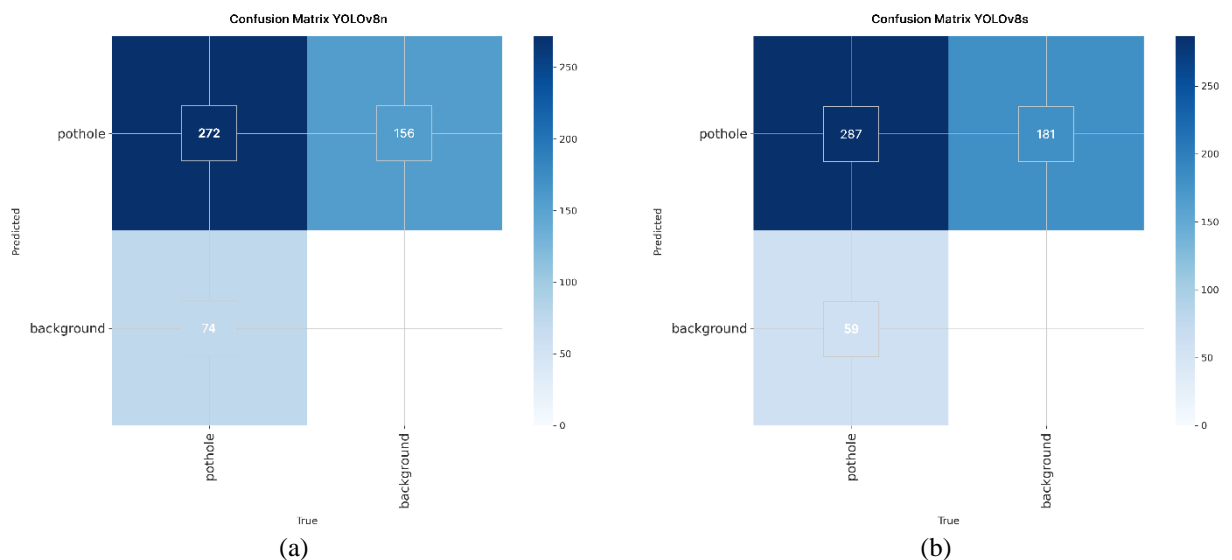


Gambar 4. Hasil Anotasi Citra Jalan Berlubang

Tahap selanjutnya dalam pra-pemrosesan adalah standarisasi ukuran gambar. Seluruh citra dalam dataset *resize* menjadi ukuran 416 x 416 piksel menggunakan parameter sesuai dengan spesifikasi model YOLOv8. Proses *resize* ini dilakukan untuk memastikan konsistensi *input* data ke model dan mempercepat proses *training*. Setelah *resize*, semua gambar dinormalisasi menggunakan normalisasi YOLO yang membagi nilai piksel dengan 255, sehingga rentang nilai piksel berada dalam interval [0, 1]. Normalisasi ini membantu stabilisasi pembelajaran model selama proses *training*. Proses pra-pemrosesan ini diterapkan pada seluruh 1.330 gambar dalam dataset, baik untuk *training* set (1.064 gambar) maupun *validation* set (266 gambar). Standarisasi ukuran gambar dan normalisasi nilai piksel memastikan bahwa semua gambar *input* memiliki format yang konsisten dan siap untuk proses pelatihan model.

3.2 Hasil Evaluasi Model

Setelah proses pelatihan dan validasi model selesai dilakukan, tahap selanjutnya adalah melakukan evaluasi mendalam terhadap performa kedua model YOLOv8s dan YOLOv8n dalam mendeteksi jalan berlubang (*pothole*). Gambar 5 menunjukkan grafik *confusion matrix* hasil pengujian kedua model deteksi jalan berlubang. *Confusion matrix* memberikan gambaran detail tentang kemampuan model dalam mengklasifikasikan objek *pothole* dan *background* dengan benar. Dari *confusion matrix* tersebut, dapat dilihat bahwa kedua model memiliki karakteristik performa yang berbeda dalam mendeteksi objek *pothole*.



Gambar 5. *Confusion matrix* YOLOv8s (a) dan YOLOv8n (b)

Berdasarkan hasil *confusion matrix* pada Gambar 5, model YOLOv8s berhasil mendeteksi 287 *pothole* dengan benar (True Positive), namun menghasilkan 59 False Negative dan 181 False Positive. Sementara itu, YOLOv8n mendeteksi 272 *pothole* dengan benar, dengan 74 False Negative dan 156 False Positive. Hasil ini menunjukkan bahwa YOLOv8s memiliki sensitivitas deteksi yang lebih tinggi dibandingkan YOLOv8n, sehingga mampu mengenali lebih banyak objek *pothole*. Namun, sensitivitas yang lebih tinggi tersebut juga menyebabkan peningkatan jumlah false positive, di mana beberapa area non-*pothole* terdeteksi sebagai *pothole*. Kesalahan deteksi tersebut umumnya terjadi pada area jalan yang memiliki tekstur kasar, retakan, bayangan, genangan air, atau perubahan intensitas permukaan yang secara visual menyerupai karakteristik *pothole*. Selain itu, kasus false negative banyak ditemukan pada *pothole* berukuran kecil atau kondisi pencahayaan yang kurang optimal sehingga objek sulit dikenali secara jelas oleh model untuk visualnya dapat dilihat pada gambar 7. Kondisi ini menunjukkan bahwa variasi visual pada permukaan jalan masih menjadi tantangan dalam proses deteksi *pothole* menggunakan pendekatan berbasis citra 2D. Nilai True Negative (TN) sebesar 0 pada kedua model menunjukkan bahwa evaluasi difokuskan pada objek *pothole* dalam skenario object detection, sehingga

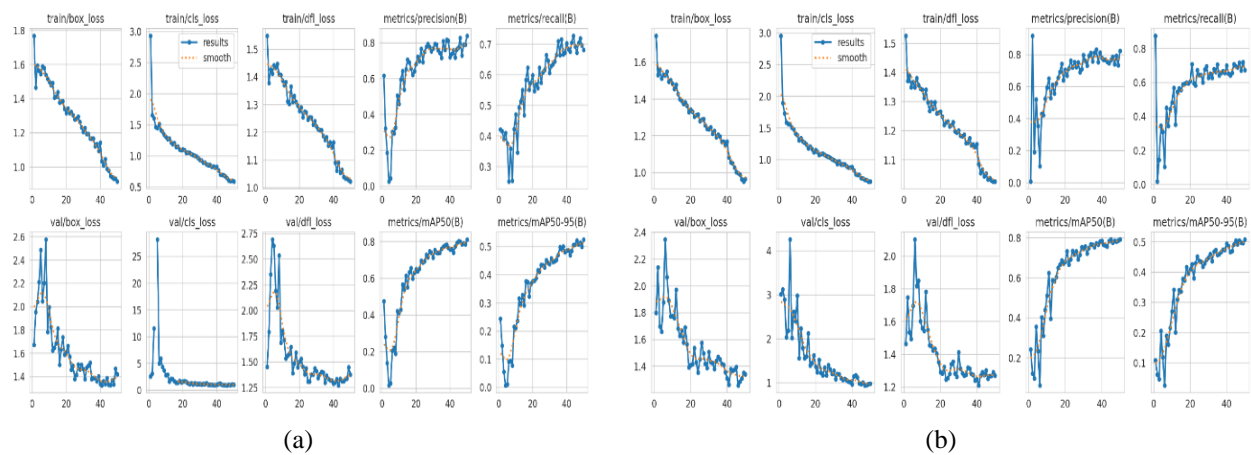
background tanpa objek tidak dihitung sebagai kelas negatif seperti pada klasifikasi biner. Secara keseluruhan, confusion matrix menunjukkan adanya trade-off antara sensitivitas deteksi dan jumlah false positive pada kedua model YOLOv8.

Untuk mengevaluasi performa kedua model secara lebih komprehensif, dilakukan pengukuran berbagai metrik evaluasi yang mencakup *Precision*, *Recall*, *mAP50*, *mAP50-95*, dan *F1-Score*. Metrik-metrik ini memberikan perspektif yang lebih menyeluruh tentang kemampuan model dalam mendeteksi objek *pothole* dengan mempertimbangkan berbagai aspek seperti akurasi prediksi, kelengkapan deteksi, dan keseimbangan antara keduanya. Tabel 4 menunjukkan hasil evaluasi akurasi kedua model yang telah diuji.

Tabel 4. Hasil Evaluasi Akurasi Model

Metrix	YOLOv8s	YOLOv8n
<i>Precision</i>	0.8243	0.8294
<i>Recall</i>	0.7023	0.6744
<i>mAP50</i>	0.8060	0.7926
<i>mAP50-95</i>	0.5595	0.5534
<i>F1-Score</i>	0.7584	0.7439

Berdasarkan hasil evaluasi pada Tabel 4, kedua model YOLOv8s dan YOLOv8n menunjukkan performa yang baik dalam mendeteksi jalan berlubang. Model YOLOv8s unggul dalam kelengkapan deteksi dengan nilai *Recall* sebesar 0,7023, lebih tinggi 2,8% dibandingkan YOLOv8n sebesar 0,6744. Pada aspek *Precision*, perbedaan kedua model sangat kecil dengan YOLOv8n mencapai 0,8294 dibandingkan YOLOv8s sebesar 0,8243. Nilai *F1-Score* juga menunjukkan keunggulan YOLOv8s dengan 0,7584 dibandingkan YOLOv8n (0,7439). Untuk metrik *mAP* yang memberikan evaluasi lebih komprehensif, YOLOv8s mencapai *mAP50* sebesar 0,8060 dan *mAP50-95* sebesar 0,5595, lebih tinggi dibandingkan YOLOv8n dengan *mAP50* sebesar 0,7926 dan *mAP50-95* sebesar 0,5534. Secara keseluruhan, YOLOv8s menunjukkan performa akurasi yang lebih unggul dalam mendeteksi dan melokalisasi jalan berlubang dibandingkan YOLOv8n. Grafik pada Gambar 6 menampilkan proses pelatihan kedua model selama 50 epoch, yang mencakup berbagai komponen loss dan matriks evaluasi. Visualisasi ini memberikan informasi penting tentang bagaimana model belajar dari data *training* dan seberapa baik kemampuan generalisasinya terhadap data validasi.



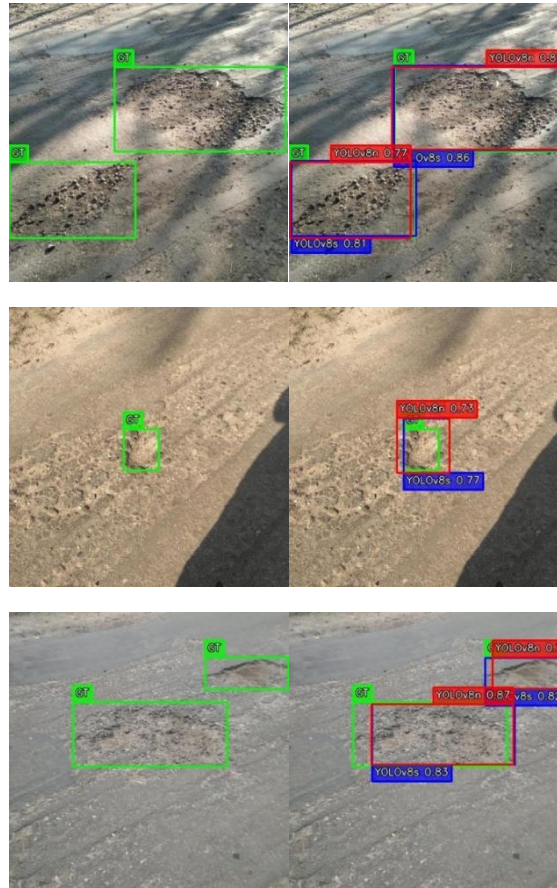
Gambar 6. Training Matrix Model YOLOv8s (a) dan YOLOv8n (b)

Grafik *training loss* menunjukkan penurunan yang konsisten pada ketiga komponen yaitu *box_loss*, *cls_loss*, dan *dfl_loss* untuk kedua model. Komponen *box_loss* yang berfungsi mengukur akurasi lokalisasi *bounding box* menurun dari nilai awal sekitar 1.8 menjadi 0.9 pada model YOLOv8s, sementara YOLOv8n dari 1.7 menjadi 0.95. Komponen *cls_loss* yang mengukur akurasi klasifikasi objek mengalami penurunan signifikan dari sekitar 3.0 hingga 0.6 pada YOLOv8s dan dari 2.9 hingga 0.7 pada YOLOv8n. Sementara itu, *dfl_loss* yang berperan dalam optimisasi distribusi lokasi objek turun dari 1.5 menjadi sekitar 1.0 pada kedua model. Penurunan yang smooth dan stabil pada semua komponen loss ini menandakan proses pembelajaran berlangsung efektif tanpa adanya fluktuasi yang ekstrem. Pada grafik *validation loss*, terlihat bahwa *val/box_loss*, *val/cls_loss*, dan *val/dfl_loss* juga mengalami penurunan meskipun dengan fluktuasi yang lebih tinggi dibandingkan *training loss*, yang merupakan hal normal karena model sedang diuji pada data yang berbeda. Yang penting adalah tren penurunan tetap konsisten dan gap antara *training loss* dengan *validation loss* tidak terlalu besar, hal ini mengindikasikan bahwa model tidak mengalami overfitting dan mampu melakukan generalisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya.

Grafik metrik *precision* dan *recall* menunjukkan peningkatan yang progresif selama proses pelatihan, di mana *precision* meningkat tajam pada 20 epoch pertama kemudian mulai stabil di kisaran 0.8 untuk YOLOv8s dan sedikit lebih tinggi untuk YOLOv8n, sementara *recall* menunjukkan peningkatan yang lebih gradual dengan nilai akhir sekitar 0.7 untuk YOLOv8s dan 0.67 untuk YOLOv8n. Grafik *mAP50* dan *mAP50-95* memperlihatkan tren peningkatan yang

konsisten sepanjang proses pelatihan, dengan $mAP50$ mencapai sekitar 0.80 dan 0.79 untuk YOLOv8s dan YOLOv8n, sementara $mAP50-95$ konvergen di sekitar 0.55 untuk kedua model. Peningkatan $mAP50-95$ yang lebih gradual dibandingkan $mAP50$ menunjukkan bahwa model memerlukan lebih banyak iterasi untuk meningkatkan akurasi lokalisasi pada threshold IoU yang lebih ketat. Kombinasi antara penurunan loss yang stabil dan peningkatan metrik evaluasi yang konsisten menunjukkan bahwa proses pelatihan berjalan optimal dan kedua model berhasil mempelajari pola deteksi *pothole* dengan baik tanpa mengalami overfitting maupun underfitting.

Gambar hasil deteksi menunjukkan bagaimana kedua model melakukan deteksi jalan berlubang. Gambar 7 menampilkan perbandingan antara ground truth (*bounding box* manual dari Roboflow dengan label GT), hasil deteksi YOLOv8s (*bounding box* berwarna biru dengan confidence score), dan hasil deteksi YOLOv8n (*bounding box* berwarna merah dengan confidence score).



Gambar 7. Hasil Deteksi Ground Truth dan Kedua Model YOLOv8

Dari visualisasi di atas, menunjukkan bahwa kedua model memiliki kemampuan yang baik dalam mendeteksi *pothole* dengan berbagai ukuran, mulai dari yang kecil hingga yang besar. Model YOLOv8s menunjukkan kecenderungan untuk menghasilkan confidence score yang lebih tinggi pada sebagian deteksi, yang mengindikasikan model ini lebih confident dalam prediksinya. Hal ini sejalan dengan karakteristik YOLOv8s yang memiliki jumlah parameter lebih banyak sehingga mampu mengekstrak fitur dengan lebih detail. Di sisi lain, YOLOv8n meskipun memiliki confidence score yang sedikit lebih rendah, tetap mampu mendeteksi objek dengan akurat dan menghasilkan *bounding box* yang presisi. Secara keseluruhan, hasil deteksi visual mengkonfirmasi bahwa kedua model YOLOv8s dan YOLOv8n mampu mendeteksi jalan berlubang dengan performa yang baik dalam berbagai kondisi, dengan YOLOv8s menunjukkan sedikit keunggulan dalam hal confidence dan kelengkapan deteksi dibandingkan dengan YOLOv8n.

Namun, pada beberapa kondisi tertentu masih ditemukan kesalahan deteksi (*false positive*), terutama pada area jalan yang memiliki bayangan, retakan, tekstur kasar, atau perubahan intensitas permukaan yang menyerupai karakteristik *pothole*. Selain itu, beberapa *false negative* terjadi pada *pothole* berukuran kecil atau kondisi pencahayaan yang kurang optimal sehingga objek tidak terdeteksi secara sempurna. Secara keseluruhan, hasil visualisasi menunjukkan bahwa YOLOv8s memiliki sensitivitas deteksi yang lebih tinggi dibandingkan YOLOv8n, meskipun dengan konsekuensi peningkatan jumlah *false positive*.

3.3 Hasil Evaluasi Efisiensi

Pada penelitian ini, dilakukan evaluasi terhadap *Inference time* dan *Frames per second* (FPS) untuk kedua model, guna menilai efisiensi kinerja kedua model dalam konteks deteksi jalan berlubang. Pengujian efisiensi komputasi dilakukan menggunakan video jalan berlubang dengan durasi 15 detik (375 frame) untuk kedua model YOLOv8s dan YOLOv8n. Setiap model menjalankan inferensi pada setiap frame video secara *real-time*, dengan waktu inferensi diukur untuk setiap frame individual. Matriks yang dikumpulkan mencakup *inference time* (waktu pemrosesan per frame), *frames per second* (FPS). Pengukuran dilakukan pada lingkungan yang sama untuk memastikan perbandingan yang adil.

Tabel 5. Spesifikasi Video Pengujian

Spesifikasi	Nilai
Resolusi	1280 x 720
Frame Rate	25.0 FPS
Durasi	15 detik
Format	MP4

Berikut pada tabel 6 merupakan hasil pengujian model Yolov8s dan Yolov8n :

Tabel 6. Hasil Pengujian Model Yolov8s Dan Yolov8n

Model	Rata-rata <i>Inference time</i>	Rata-rata FPS	Total Frame
YOLOv8s	13.44 ms	74.43	375
YOLOv8n	11.96 ms	83.63	375

Hasil pengujian menunjukkan perbedaan signifikan dalam hal kecepatan pemrosesan antara kedua model. Model YOLOv8n mencapai rata-rata *inference time* sebesar 11.96 ms per frame, yang lebih cepat 11.60% dibandingkan dengan YOLOv8s yang mencapai 13.44 ms per frame. Perbedaan ini tercermin dalam nilai FPS, di mana YOLOv8n mampu memproses 83.63 frame per detik, lebih tinggi 12.36% dibandingkan YOLOv8s yang mencapai 74.43 FPS, sehingga model YOLOv8n lebih cocok untuk aplikasi yang memprioritaskan kecepatan deteksi. Sebaliknya, meskipun YOLOv8s memiliki performa yang sedikit lebih lambat, model ini tetap mampu melakukan deteksi *real-time* dan menghasilkan akurasi yang lebih tinggi. Visualisasi pada Gambar 8 menunjukkan perbandingan performa deteksi kedua model.



Gambar 8. Pengujian Deteksi Model YOLOv8s dan YOLOv8n

Model YOLOv8n juga menunjukkan keunggulan signifikan dalam hal ukuran file dengan hanya 5.93 MB, yang 261.64% lebih kecil dibandingkan YOLOv8s sebesar 21.45 MB. Perbedaan ukuran model ini berbanding lurus dengan jumlah parameter, di mana YOLOv8n memiliki 3.1 juta parameter, jauh lebih sedikit dari YOLOv8s yang memiliki 11.1 juta parameter. Ukuran model yang lebih kecil menjadikan YOLOv8n lebih mudah untuk disimpan, didistribusikan, dan dijalankan pada perangkat dengan keterbatasan memori, terutama untuk aplikasi edge computing atau sistem embedded. Meskipun demikian, perbedaan computational complexity (28.6 GFLOPs untuk YOLOv8s dan 8.2 GFLOPs untuk YOLOv8n) mencerminkan efisiensi komputasi yang lebih baik pada model YOLOv8n, yang konsisten dengan hasil pengukuran *inference time* dan FPS yang diperoleh dari pengujian video.

Tabel 7. Spesifikasi Model Yolov8s Dan Yolov8n

Spesifikasi	YOLOv8s	YOLOv8n
Model Size	21.45 mb	5.93 mb
GFLOPs	28.6	8.2

4. KESIMPULAN

Penelitian ini menunjukkan bahwa kedua varian lightweight YOLOv8, yaitu YOLOv8s dan YOLOv8n, memiliki karakteristik performa yang berbeda dalam tugas deteksi jalan berlubang berbasis citra 2D. YOLOv8s memberikan kemampuan deteksi yang lebih baik dengan sensitivitas dan akurasi yang lebih tinggi dibandingkan YOLOv8n,

menunjukkan bahwa model dengan jumlah parameter lebih besar mampu mengekstraksi fitur pothole secara lebih detail pada berbagai kondisi permukaan jalan. Namun, peningkatan performa tersebut diikuti dengan kebutuhan komputasi yang lebih tinggi, baik dari sisi waktu inferensi maupun ukuran model. Sebaliknya, YOLOv8n menawarkan efisiensi komputasi yang lebih baik dengan ukuran model yang lebih ringan dan kecepatan inferensi yang lebih tinggi. Hal ini menunjukkan bahwa YOLOv8n memiliki potensi implementasi pada perangkat dengan keterbatasan sumber daya komputasi. Meskipun demikian, nilai mAP50-95 yang masih berada pada kisaran 0,55 menunjukkan bahwa kedua model masih memiliki keterbatasan dalam menghasilkan deteksi yang konsisten pada berbagai kondisi kompleks di lapangan, khususnya pada objek dengan tekstur tidak jelas, pencahayaan rendah, bayangan, maupun permukaan jalan yang menyerupai pothole. Hasil penelitian juga menunjukkan adanya trade-off antara sensitivitas deteksi dan jumlah false positive. YOLOv8s cenderung lebih sensitif dalam mendeteksi pothole, tetapi menghasilkan false positive yang lebih tinggi, sedangkan YOLOv8n lebih efisien namun cenderung melewatkan beberapa objek pothole. Temuan ini mengindikasikan bahwa penerapan model deteksi pothole berbasis YOLOv8 masih memerlukan pengembangan lebih lanjut sebelum digunakan pada skenario infrastruktur kritis secara penuh. Untuk penelitian selanjutnya, pengembangan dapat difokuskan pada peningkatan robustness model melalui evaluasi pada dataset lintas lingkungan, penerapan validasi silang, eksplorasi teknik augmentasi yang lebih spesifik terhadap karakteristik pothole, serta optimasi arsitektur dan threshold deteksi untuk menekan false positive tanpa mengurangi sensitivitas model secara signifikan. Selain itu, integrasi data spasial atau sensor kedalaman dapat menjadi pendekatan potensial untuk meningkatkan reliabilitas deteksi pada kondisi jalan yang kompleks.

REFERENCES

- [1] S. Sudirman, "Analisis Tingkat Kerusakan Jalan: Studi Kasus: Jalan Trans Sulawesi, Kecamatan Telluwanua, Km 10 hingga Km 12 Kota Palopo," *J. Ilm. Ecosyst.*, vol. 23, no. 1, pp. 40–51, Apr. 2023, doi: 10.35965/eco.v23i1.2502.
- [2] R. A. Gumelar and A. Susetyaningsih, "Pengaruh Kerusakan Jalan Terhadap Kenyamanan Pengguna Jalan di Jalan Raya," *J. Konstr.*, vol. 21, no. 2, pp. 265–274, Oct. 2023, doi: 10.33364/konstruksi/v.21-2.1416.
- [3] Y. Zanevych, V. Yovbak, O. Basystiuk, N. Shakhovska, S. Fedushko, and S. Argyroudis, "Evaluation of Pothole Detection Performance Using Deep Learning Models Under Low-Light Conditions," *Sustainability*, vol. 16, no. 24, p. 10964, Dec. 2024, doi: 10.3390/su162410964.
- [4] R. Thakur, B. Bhumika, P. Kumar, and P. Thakur, "YOLOv8-based Pothole Detection: A Real-time Approach for Road Infrastructure Monitoring," *SSRN Electron. J.*, 2025, doi: 10.2139/ssrn.5089189.
- [5] Esadhipa Raif Syihabuddin, Widia Angela, Muhammad Naufal, and Ricardus Anggi Pramunendar, "PENGENALAN PLAT NOMOR KENDARAAN DENGAN YOLOv8 DAN PADDLEOCR," *J. Comput. Sci. Technol. JCS-TECH*, vol. 5, no. 2, pp. 66–72, Nov. 2025, doi: 10.54840/jestech.v5i2.374.
- [6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020, *arXiv*. doi: 10.48550/ARXIV.2004.10934.
- [7] M. Yurdakul and Ş. Tasdemir, "An Enhanced YOLOv8 Model for Real-Time and Accurate Pothole Detection and Measurement," 2025, *arXiv*. doi: 10.48550/ARXIV.2505.04207.
- [8] X. Wang, H. Gao, Z. Jia, and Z. Li, "BL-YOLOv8: An Improved Road Defect Detection Model Based on YOLOv8," *Sensors*, vol. 23, no. 20, p. 8361, Oct. 2023, doi: 10.3390/s23208361.
- [9] San Gabriel Vanness Kenrick Erwi and Hafiz Irsyad, "Implementasi Deteksi Objek Pada Jalan Rusak Menggunakan Metode YOLOv8," *Bul. Ilm. Inform. Teknol.*, vol. 3, no. 1, pp. 1–9, Oct. 2024, doi: 10.58369/biit.v3i1.76.
- [10] A. Tripathy, T. Rajalakshmi, S. K. R., and S. V. Suryakala, "Image-Based Pothole Detection System Using YoloV8 Algorithm," in *2024 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI)*, Chennai, India: IEEE, Apr. 2024, pp. 1–5. doi: 10.1109/RAEEUCCI61380.2024.10547946.
- [11] L. Sakinah, E. Haryatmi, and T. A. Riyadi, "Implementasi Algoritma Yolo Untuk Mendeteksi Jalan Berlubang dan Retak," *JITSI J. Ilm. Teknol. Sist. Inf.*, vol. 6, no. 3, Sep. 2025, doi: 10.62527/jitsi.6.3.488.
- [12] J. Zhong, D. Kong, Y. Wei, and B. Pan, "YOLOv8 and point cloud fusion for enhanced road pothole detection and quantification," *Sci. Rep.*, vol. 15, no. 1, p. 11260, Apr. 2025, doi: 10.1038/s41598-025-94993-0.
- [13] RUCHI, "Pothole detection Dataset." Roboflow Universe, 2022. [Online]. Available: <https://universe.roboflow.com/ruchi/pothole-detection-sfxgi>
- [14] Jocher, G., Chaurasia, A., & Qiu, J, "Ultralytics YOLOv8 [Computer software]," 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [15] R. F. Puspita, M. Naufal, and F. Al Zami, "Improving YOLO Performance with Advanced Data Augmentation for Soccer Object Detection," *J. Appl. Inform. Comput.*, vol. 9, no. 6, pp. 3601–3611, Dec. 2025, doi: 10.30871/jaic.v9i6.11256.
- [16] "what is yolov8." [Online]. Available: <https://yolov8.org/what-is-yolov8/>
- [17] F. Alzami *et al.*, "Bayesian-Optimized CLAHE for Enhanced Drowsiness Detection in Low-Light Conditions Using Time-Distributed MobileNetV2-GRU Architecture," *Stat. Optim. Inf. Comput.*, vol. 15, no. 1, pp. 274–294, Oct. 2025, doi: 10.19139/soic-2310-5070-3024.
- [18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
- [19] K. Gorro, E. Ranolo, L. Roble, and R. N. Santillan, "Road Pothole Detection Using YOLOv8 with Image Augmentation," *J. Image Graph.*, vol. 12, no. 4, 2024, doi: 10.18178/joig.12.4.417-426.
- [20] T. Taufiqurrahman, A. P. Hadi, and R. E. Siregar, "Evaluasi Performa YOLOv8 Dalam Deteksi Objek Di Depan Kendaraan Dengan Variasi Kondisi Lingkungan," *J. Minfo Polgan*, vol. 13, no. 2, pp. 1755–1773, Nov. 2024, doi: 10.33395/jmp.v13i2.14228.

- [21] R. Padilla, S. L. Netto, and E. A. B. Da Silva, “A Survey on Performance Metrics for Object-Detection Algorithms,” in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Niterói, Brazil: IEEE, Jul. 2020, pp. 237–242. doi: 10.1109/IWSSIP48289.2020.9145130.
- [22] J. Chang, Z. Chen, and E. Xia, “Improved YOLOv8 Method for Multi-scale Pothole Detection,” in *Advanced Intelligent Computing Technology and Applications*, vol. 14872, D.-S. Huang, Y. Pan, and Q. Zhang, Eds., in *Lecture Notes in Computer Science*, vol. 14872. , Singapore: Springer Nature Singapore, 2024, pp. 383–395. doi: 10.1007/978-981-97-5612-4_33.