

PENERAPAN MODEL YOLO UNTUK DETEKSI KERUSAKAN JALAN BERDASARKAN CITRA VISUAL**Amanda Iza Sofiani***

Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi, Jambi, Indonesia
e-mail: amanda.iza.sofiani@gmail.com

Nazwa Eka Hervy

Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi, Jambi, Indonesia
e-mail: nazwaa.eka20@gmail.com

Fitri Dwi Lestari

Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi, Jambi, Indonesia
e-mail: fitridwil045@gmail.com

M. Rizky Ardiansyah Putra

Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi, Jambi, Indonesia
e-mail: rizkyjambi859@gmail.com

Mutia Fadhila Putri

Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi, Jambi, Indonesia
e-mail: mutia.fadhila@unja.ac.id

Ulfa Khaira

Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi, Jambi, Indonesia
e-mail: ulfa.ilkom@gmail.com

Pradita Eko Prasetyo Utomo

Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi, Jambi, Indonesia
e-mail: pradita.eko@unja.ac.id

Abstrak

Kerusakan jalan seperti retak dan lubang menjadi permasalahan infrastruktur yang mengganggu keselamatan dan kelancaran lalu lintas. Metode identifikasi manual yang selama ini digunakan dinilai kurang efisien. Penelitian ini bertujuan untuk menerapkan model YOLOv8 dalam mendekripsi kerusakan jalan secara otomatis melalui citra visual. Metode yang digunakan adalah pendekatan kuantitatif eksperimental dengan pelatihan model deep learning menggunakan 200 citra yang dianotasi secara manual. Data dibagi ke dalam skema pelatihan, validasi, dan pengujian (70:20:10). Hasil pelatihan menunjukkan bahwa YOLOv8 mampu mendekripsi dua jenis kerusakan utama, yaitu retak dan berlubang, dengan performa yang cukup baik. Model menunjukkan peningkatan nilai precision hingga 0.4 dan recall lebih dari 0.3. Sementara itu, nilai mAP50 mencapai sekitar 0.27, dan mAP50-95 lebih dari 0.1. Temuan ini mengindikasikan bahwa model cukup andal dalam mengenali kerusakan jalan dengan akurasi memadai, serta memiliki potensi untuk diintegrasikan dalam sistem monitoring infrastruktur secara real-time.

Kata Kunci: citra visual, deep learning, deteksi kerusakan jalan, object detection, YOLOv8

Abstract

Road damage such as cracks and potholes poses a threat to traffic safety and efficiency. Traditional manual inspection methods are often inefficient. This study aims to implement the YOLOv8 model for automated detection of road damage using visual imagery. The method employed is an experimental quantitative approach, involving deep learning model training using 200 manually annotated images. The dataset is split into training, validation, and testing sets (70:20:10). The training results show that YOLOv8 effectively detects two major types of road damage: cracks and potholes. The model achieved a precision of up to 0.4 and recall exceeding 0.3. Moreover, the mAP50 value reached

approximately 0.27, while the mAP50-95 exceeded 0.1. These findings indicate that the model is reliable in identifying road damage with sufficient accuracy and has the potential to be integrated into real-time infrastructure monitoring systems.

Keywords: visual imagery, deep learning, road damage detection, object detection, YOLOv8

PENDAHULUAN

Kerusakan jalan merupakan permasalahan infrastruktur yang masih sering dijumpai di berbagai wilayah, baik di perkotaan maupun pedesaan. Kondisi jalan yang berlubang, retak, atau tidak rata dapat membahayakan keselamatan pengguna jalan dan mengganggu kelancaran arus lalu lintas. Secara sosial, hal ini berdampak pada meningkatnya risiko kecelakaan dan menurunnya kenyamanan masyarakat dalam berkendara. Dari sisi administratif, proses identifikasi dan perbaikan kerusakan jalan kerap kali lambat karena masih dilakukan secara manual, yang bergantung pada laporan masyarakat atau inspeksi lapangan yang memakan waktu dan tenaga. Maka dari itu, diperlukan solusi berbasis teknologi yang mampu mendeteksi kerusakan jalan secara otomatis dan efisien. Menurut (Pramesty, 2018) menunjukkan bahwa penerapan metode YOLO berbasis citra digital dapat mendeteksi jenis kerusakan seperti lubang, retak garis, dan retak non-garis secara cepat dan akurat, dengan tingkat akurasi klasifikasi mencapai 99,25%.

Berbagai studi sebelumnya telah mengeksplorasi metode deteksi kerusakan jalan berbasis citra visual menggunakan pendekatan machine learning dan deep learning. Penelitian oleh (Utomo, 2020) mengevaluasi performa algoritma klasifikasi seperti *Naive Bayes*, *Nearest Neighbor*, *SVM*, dan *Decision Tree* untuk membedakan citra jalan rusak dan tidak rusak berdasarkan ekstraksi ciri histogram. Namun, pendekatan dua tahap seperti *Faster R-CNN* cenderung memiliki waktu inferensi yang lebih lambat. Alternatif yang lebih efisien adalah penggunaan YOLO (*You Only Look Once*), sebuah model deteksi objek satu tahap (*one-stage detector*) yang mampu melakukan deteksi secara real-time. Di sisi lain, (Pramesty, 2018) memanfaatkan YOLO yang mengintegrasikan CNN untuk mendeteksi berbagai jenis kerusakan jalan secara *real-time*, membuktikan keunggulannya dalam hal kecepatan dan efektivitas deteksi objek pada citra jalan. Namun, model deteksi dua tahap seperti *Faster R-CNN* yang digunakan dalam penelitian sebelumnya

cenderung memerlukan waktu inferensi yang lebih lama, sehingga tidak optimal untuk aplikasi real-time

Berdasarkan permasalahan dan temuan dari penelitian sebelumnya, tujuan dari penelitian ini adalah untuk menerapkan model YOLOv8 dalam proses deteksi otomatis kerusakan jalan berdasarkan citra visual. Penelitian oleh (Erwi, 2024) menunjukkan bahwa implementasi YOLOv8 dalam deteksi jalan rusak memberikan hasil yang menjanjikan meskipun masih memiliki tantangan dalam meningkatkan akurasi yang optimal. Penelitian juga bertujuan untuk menguji generalisasi model terhadap data nyata dengan kondisi visual yang bervariasi, seperti pencahayaan, sudut pandang, dan cuaca, guna menilai efektivitas dan ketahanan model dalam skenario dunia nyata.

Penerapan YOLOv8 dalam deteksi kerusakan jalan diyakini mampu menjadi solusi teknologi yang praktis untuk menggantikan metode inspeksi manual yang lambat dan subjektif. Berdasarkan kemampuan YOLO untuk melakukan deteksi objek dalam waktu nyata dan dengan tingkat akurasi tinggi, penelitian ini diharapkan dapat menghasilkan sistem yang mampu diintegrasikan dengan kamera kendaraan atau sistem monitoring otomatis. Hasil penelitian ini diharapkan dapat memberikan kontribusi nyata bagi pengembangan sistem transportasi cerdas, khususnya dalam pengelolaan infrastruktur jalan, serta mendorong pemanfaatan teknologi deep learning dalam pemecahan masalah sosial yang berdampak luas.

KAJIAN TEORI

KECERDASAN BUATAN (ARTIFICIAL INTELLIGENCE)

Kecerdasan buatan (*Artificial Intelligence/AI*) adalah cabang ilmu komputer yang memungkinkan sistem melakukan tugas-tugas kompleks seperti pengenalan pola dan pengambilan keputusan secara otomatis. Dalam bidang pengolahan citra digital, AI digunakan untuk mendeteksi objek berdasarkan karakteristik visual dalam gambar.

Pada konteks deteksi kerusakan jalan, seperti lubang dan retakan, AI dapat dimanfaatkan untuk menggantikan metode manual yang cenderung

lambat dan subjektif. Mahar et al. (2025) menyatakan bahwa penerapan AI berbasis algoritma YOLOv8 mampu mendeteksi lubang jalan secara *real-time* dengan akurasi yang memadai, sehingga menjadi solusi efektif untuk meningkatkan efisiensi dan objektivitas dalam pemantauan infrastruktur.

DEEP LEARNING

Deep learning adalah subbidang dari *machine learning* yang menggunakan jaringan saraf tiruan (*artificial neural networks*) dengan banyak lapisan untuk memproses dan menganalisis data yang kompleks. *Deep learning* merupakan pengembangan dari jaringan saraf tiruan (*neural network*) multi lapisan untuk memberikan ketepatan tugas seperti deteksi objek dan pengenalan suara (Raup et al., 2022).

Jaringan saraf terdiri dari lapisan input, tersembunyi, dan output yang memproses data melalui *forward propagation*. Perbedaan antara prediksi dan nilai target dihitung menggunakan fungsi loss, dan bobot jaringan diperbarui melalui *backpropagation* menggunakan algoritma seperti SGD, Adam, dan RMSprop. *Convolutional Neural Networks* (CNN) merupakan salah satu algoritma yang digunakan dalam *deep learning* untuk memproses gambar atau mendeteksi objek (Maszeri et.al, 2024).

CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolutional Neural Network (CNN) merupakan arsitektur jaringan saraf tiruan yang dirancang khusus untuk mengolah data berbentuk *grid* dua dimensi, seperti citra digital. CNN banyak digunakan dalam tugas pengenalan pola visual karena kemampuannya dalam mengekstraksi fitur lokal melalui operasi konvolusi dan mempelajari representasi tingkat tinggi dari data gambar (Tilasefana & Putra, 2023).

Terdapat empat lapisan utama arsitektur dasar CNN (Mishra, 2020):

1. *Convolution Layer*. Operasi konvolusi ini menghasilkan *feature map* yang mencerminkan keberadaan pola tertentu pada posisi tertentu, dirumuskan dengan:

$$(I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot K(m, n)$$

2. *Activation Function* (ReLU), diterapkan untuk memperkenalkan non-linearitas ke dalam model. Fungsi ReLU didefinisikan sebagai:

$$f(x) = \max(0, x)$$

3. *Pooling Layer*, digunakan untuk mengurangi dimensi spasial dari feature map dan memperkuat fitur dominan.
4. *Fully Connected Layer* (FC Layer), adalah lapisan akhir CNN yang menghubungkan seluruh neuron untuk menghasilkan prediksi akhir, digunakan untuk klasifikasi.

OBJECT DETECTION

Object detection merupakan cabang utama dalam computer vision yang bertujuan mengidentifikasi dan melokalisasi objek dalam citra atau video, menghasilkan *bounding box* dan label kelas (Irfansyah et al., 2024). Teknik ini berbeda dengan *image classification* yang hanya mengklasifikasi seluruh citra, dan *semantic segmentation* yang memberi label per piksel. *Instance segmentation* merupakan pendekatan lebih kompleks yang memisah.

YOLO (*You Only Look Once*) menjadi pilihan utama dalam penelitian terbaru karena kemampuannya menyeimbangkan kecepatan dan akurasi. YOLOv8 mencapai akurasi 43,98% pada dataset 26.336 citra jalan dengan pembagian 80:10:10 (pelatihan:validasi:uji) (Erwi & Irsyad, 2024).

MODEL YOLO (YOU ONLY LOOK ONCE)

YOLO (*You Only Look Once*) adalah salah satu algoritma deteksi objek paling populer yang menggunakan pendekatan *one-stage detection*, di mana proses prediksi lokasi dan klasifikasi objek dilakukan secara simultan dalam satu tahap. Pendekatan ini membuat YOLO unggul dalam hal kecepatan dan efisiensi, terutama dalam aplikasi *real-time* seperti deteksi kerusakan jalan (Firgiawan et al., 2024).

YOLO pertama kali diperkenalkan pada tahun 2016 (YOLOv1) dan terus berkembang hingga versi terbaru YOLOv8. Versi-versi awal YOLO (v1-v3) fokus pada peningkatan akurasi dan efisiensi. YOLOv4 dan YOLOv5 menghadirkan peningkatan signifikan dari sisi *training pipeline* dan fleksibilitas arsitektur. YOLOv8 sebagai versi terbaru

menawarkan model yang lebih ringan, mendukung task tambahan (seperti *instance segmentation* dan *classification*), serta kompatibel dengan framework Ultralytics untuk pelatihan yang lebih efisien (Reddy et al., 2024).

YOLO membagi citra input menjadi beberapa *grid cell*. Setiap *grid* bertanggung jawab untuk memprediksi koordinat lokasi objek (x, y, lebar, tinggi menggunakan *bounding box*), probabilitas keberadaan objek dalam *bounding box* dan tingkat keyakinan *bounding box* tersebut benar menggunakan *confidence score*. YOLO menggunakan kombinasi dari tiga jenis *loss*:

$$\text{Loss} = \text{Localization Loss} + \text{Confidence Loss} + \text{Classification Loss}$$

Terdapat sejumlah perhitungan matematis penting yang mendasari proses deteksi dan klasifikasi objek.

1. *Intersection over Union* (IoU), digunakan untuk mengukur seberapa besar prediksi *bounding box* tumpang tindih dengan *ground truth*:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

2. *Confidence Score*, menggabungkan probabilitas keberadaan objek dan kualitas prediksi *bounding box*:

$$\text{Confidence} = Pr(\text{Object}) \times IoU$$

3. *Class Score* untuk mendapatkan skor klasifikasi akhir dari suatu objek:

$$\text{Class Score} = \text{Confidence} \times P(\text{class}_i)$$

PENGOLAHAN CITRA DIGITAL

Menurut Chaky & Dey (2018), pengolahan citra digital (*digital image processing*) merupakan teknik pemrosesan informasi visual berbasis komputer yang bertujuan untuk meningkatkan kualitas gambar atau mengekstraksi informasi penting dari gambar tersebut. Sebelum citra digunakan dalam pelatihan model, dilakukan tahap pra-pemrosesan (*preprocessing*) yang bertujuan untuk menstandarkan data dan meningkatkan kinerja model deteksi dengan normalisasi atau pun dalam kasus ini menggunakan *manual labelling*.

Citra digital umumnya disimpan dalam format RGB (*Red, Green, Blue*) yang menyusun warna berdasarkan tiga kanal warna utama. Namun, dalam beberapa kasus, model juga dapat dilatih menggunakan format *grayscale*, terutama ketika

informasi warna tidak terlalu penting atau untuk mengurangi kompleksitas data (Abdiansyah et al., 2021).

EVALUASI MODEL

Evaluasi kinerja model deteksi objek bertujuan untuk mengukur seberapa baik model dalam mengidentifikasi dan mengklasifikasikan objek pada citra.

Tiga metrik dasar dalam evaluasi klasifikasi adalah *precision*, *recall*, dan *F1-score* (Padilla et al., 2021). Ketiganya didasarkan pada hasil prediksi model yang diklasifikasikan ke dalam empat kategori: *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN).

- *Precision*, mengukur proporsi prediksi positif yang benar:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- *Recall*, mengukur proporsi objek yang relevan berhasil terdeteksi:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- *F1-score*, rata-rata harmonik dari *precision* dan *recall*, digunakan untuk menyeimbangkan keduanya:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Dalam deteksi objek, *mean Average Precision* (mAP) penting digunakan (Hidayat et al., 2023). Metrik ini mengukur akurasi model berdasarkan rata-rata *precision* di berbagai ambang batas IoU (*Intersection over Union*):

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

METODE

RANCANGAN PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif eksperimental dengan metode pemodelan deep learning untuk deteksi kerusakan jalan berbasis citra visual. Penelitian dilakukan dengan menerapkan model YOLOv8 sebagai arsitektur deteksi objek utama. Model dilatih dan diuji pada dataset citra jalan yang telah diberi anotasi *bounding box* berdasarkan jenis kerusakan (lubang, retak garis, dan retak non-garis). Proses pelatihan, validasi, dan pengujian model dilakukan secara

bertahap menggunakan pembagian data dengan skema 80% untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian.

POPULASI DAN SAMPEL

Populasi dalam penelitian ini adalah citra jalan yang diambil dari lingkungan perkotaan dan pedesaan dengan kondisi jalan yang bervariasi. Sampel berupa dataset citra visual jalan rusak yang bersumber dari dua jenis data: dataset publik (*open-source*) dan data primer hasil pengambilan gambar secara langsung di lapangan. Citra dipilih berdasarkan variasi kerusakan, kondisi pencahayaan, sudut pandang, serta latar belakang yang beragam guna menguji kemampuan generalisasi model.

TEKNIK PENGUMPULAN DATA

Data dalam penelitian ini dikumpulkan melalui metode pengambilan gambar dari internet. Sumber data terdiri dari berbagai situs web, media berita, blog teknis, dan platform berbagi gambar seperti Google Images dan Flickr, yang menyediakan citra jalan dengan berbagai jenis kerusakan.

Setelah dikumpulkan, citra-citra tersebut dilakukan proses anotasi untuk menandai area kerusakan dengan *bounding box*. Anotasi disimpan dalam format YOLO (.txt) yang sesuai dengan kebutuhan model YOLOv8.

TEKNIK ANALISIS DATA

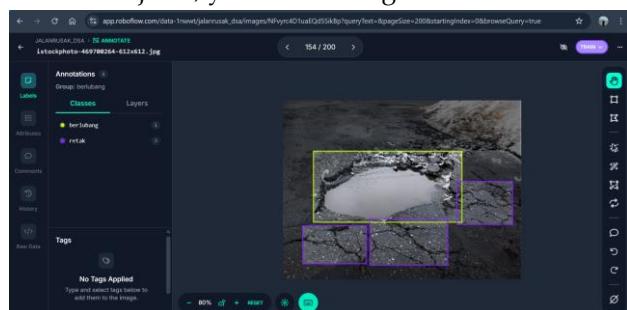
Data dianalisis melalui tahapan sebagai berikut:

- **Pra-Pemrosesan Citra:** Melalui manual labelling.
- **Pelatihan Model:** Model YOLOv8 dilatih menggunakan data yang telah diproses dengan konfigurasi epoch sebanyak 100, batch size 16, dan learning rate default.
- **Evaluasi Kinerja:** Model dievaluasi menggunakan metrik *precision*, *recall*, *F1-score*, dan *mean Average Precision (mAP)* pada berbagai nilai ambang *Intersection over Union (IoU)*.
- **Pengujian Generalisasi:** Untuk menilai ketahanan model, dilakukan pengujian terhadap data baru dari lingkungan berbeda dengan kondisi pencahayaan dan sudut pandang bervariasi.

HASIL DAN PEMBAHASAN

DATA PREPROCESSING

Tahap awal dalam penerapan model deteksi kerusakan jalan dimulai dengan proses *preprocessing* data, yang bertujuan untuk mempersiapkan dataset sebelum memasuki fase pelatihan model. Sebanyak 200 citra visual digunakan dan melalui proses pelabelan secara manual menggunakan platform Roboflow dengan metode *object detection*. Dalam proses anotasi, diterapkan dua kelas utama kerusakan jalan, yaitu berlubang dan retak.

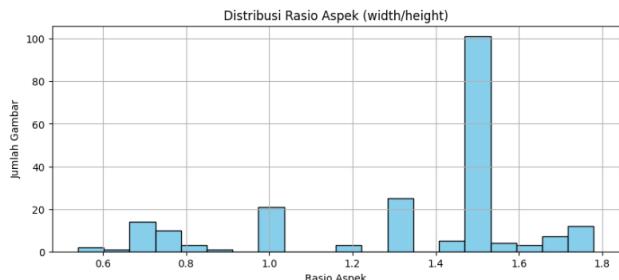


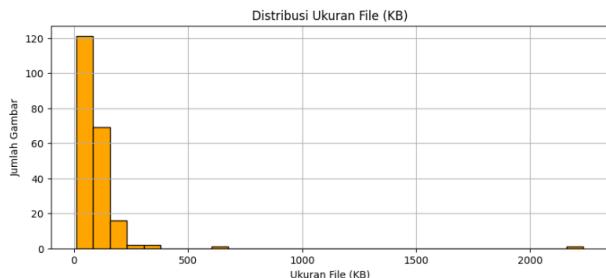
Gambar 1. Contoh Hasil Anotasi Data

Setelah proses pelabelan selesai, dataset dibagi menjadi 3 subset dengan proporsi 70% untuk data *training*, 20% untuk data *validation*, dan 10% untuk data *testing* guna mendukung proses pelatihan dan evaluasi model secara optimal. Seluruh data yang telah diproses kemudian diekspor dalam format YOLOv8 agar sesuai dengan arsitektur model yang digunakan dalam sistem deteksi ini.

EXPLORATORY DATA ANALYSIS

Tahap *Exploratory Data Analysis (EDA)* dilakukan untuk memahami karakteristik dasar dari data citra sebelum dimasukkan ke dalam proses pelatihan model. Analisis dilakukan terhadap beberapa aspek penting, seperti resolusi gambar, rasio aspek, dan ukuran file.





Gambar 2. Distribusi Rasio dan Ukuran Gambar dalam Dataset

Hasil analisis menunjukkan bahwa resolusi gambar yang paling sering ditemukan dalam dataset adalah 612×408 piksel, yang kemungkinan merupakan hasil dari proses penyesuaian ukuran otomatis atau merupakan preferensi standar dalam persiapan data. Resolusi ini dinilai cukup optimal untuk menjaga keseimbangan antara kualitas visual dan efisiensi pemrosesan. Selain itu, rasio aspek gambar yang dominan berada di kisaran 1,5 atau 3:2, mengindikasikan konsistensi proporsi lebar dan tinggi gambar yang dapat mempermudah proses pelatihan model. Dari sisi ukuran file, mayoritas gambar memiliki ukuran di bawah 200 KB, yang menunjukkan bahwa dataset ini relatif ringan dan efisien untuk digunakan dalam pemrosesan visual berskala besar tanpa membebani sumber daya komputasi secara signifikan.

HYPERPARAMETER TUNING

Tahap *Hyperparameter Tuning* dilakukan untuk mengoptimalkan kinerja model YOLO dalam mendeteksi jenis kerusakan jalan seperti retak dan berlubang. Proses ini bertujuan untuk menemukan kombinasi nilai *hyperparameter* terbaik yang dapat meningkatkan akurasi prediksi pada data validasi, sekaligus menjaga stabilitas selama pelatihan.

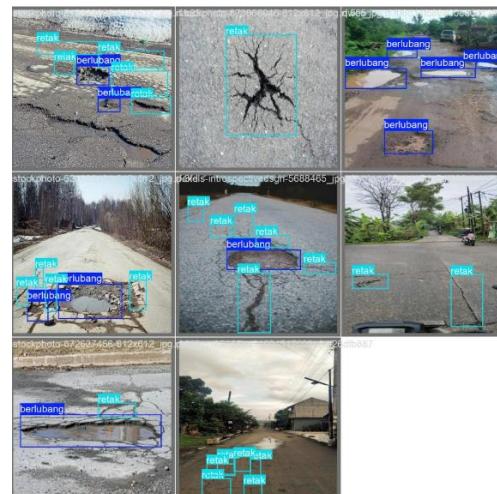
Penentuan jumlah *epochs* dilakukan secara bertahap untuk menghindari overfitting. Nilai *batch size* diuji dengan beberapa variasi untuk menyeimbangkan antara stabilitas pelatihan dan efisiensi komputasi. Sementara itu, *learning rate* disesuaikan secara hati-hati karena berpengaruh langsung terhadap kecepatan dan konvergensi model selama pelatihan. Ukuran *input image* juga turut diuji, karena ukuran input yang lebih besar dapat meningkatkan detail fitur yang ditangkap, tetapi memerlukan waktu pelatihan yang lebih lama.

Eksperimen dilakukan dengan menyesuaikan berbagai parameter penting. Nilai *learning rate* ($lr0$) disetel sebesar **0.001** untuk memastikan proses pembelajaran berlangsung secara stabil. *Momentum* ditetapkan pada nilai **0.8**, yang membantu mempercepat konvergensi dengan tetap menjaga kestabilan gradien. Untuk mengurangi risiko overfitting, *weight decay* disetel pada **0.0001**.

Pada sisi *loss function*, parameter *box* diatur ke **0.05** dan *cls* ke **0.7** untuk mengontrol kontribusi kesalahan dalam prediksi posisi dan kelas objek secara seimbang. Selain itu, dilakukan augmentasi data untuk meningkatkan kemampuan generalisasi model. Augmentasi ini meliputi rotasi dengan *degrees* sebesar **5**, translasi gambar (*translate*) sebesar **0.1**, serta pembalikan horizontal (*flip left-right*) dengan probabilitas **0.5**.

Model dilatih selama **100 epochs** dengan ukuran gambar (*imgsz*) sebesar **640 piksel** dan *batch size* sebesar **16**. Kombinasi parameter tersebut dipilih berdasarkan evaluasi terhadap performa model pada data validasi, dengan mempertimbangkan metrik seperti *precision*, *recall*, dan *mAP* (*mean Average Precision*). Hasil tuning ini menunjukkan bahwa model mampu mendeteksi kerusakan jalan secara efektif dengan tingkat presisi yang tinggi.

OBJECT DETECTION & CLASSIFICATION



Gambar 3. Visualisasi Hasil Deteksi Kerusakan Jalan oleh Model YOLOv8

Model deteksi objek berbasis YOLO (*You Only Look Once*) yang telah dilatih untuk mendeteksi kerusakan jalan menunjukkan performa yang baik dalam mengidentifikasi dua jenis kerusakan utama, yaitu **retak** dan **berlubang**. Berdasarkan hasil

visualisasi pada gambar di atas, model mampu menghasilkan prediksi dengan *bounding box* yang cukup akurat, serta memberikan label klasifikasi yang sesuai.

AKURASI DETEKSI OBJEK

Model berhasil mengenali dan membedakan antara **keretakan (retak)** dan **lubang jalan (berlubang)** di berbagai kondisi pencahayaan dan latar belakang. Hal ini terlihat dari:

- Banyaknya area yang ditandai dengan label "**retak**" berwarna cyan dan "**berlubang**" berwarna biru.
- Hasil deteksi pada beberapa gambar menunjukkan bahwa model mampu mengenali **lebih dari satu kerusakan** dalam satu citra, baik itu retakan berganda maupun kombinasi antara lubang dan retakan.

PRESISI DETEKSI

- Deteksi keretakan cukup presisi, terutama pada gambar dengan latar permukaan jalan yang jelas dan kontras yang tinggi. *Bounding box* menyelimuti area retak dengan baik tanpa banyak kesalahan deteksi.
- Deteksi lubang juga cukup akurat pada gambar dengan pencahayaan baik, meskipun pada beberapa gambar lubang dengan tepian tidak terlalu kontras masih bisa dideteksi dengan benar.

KELEBIHAN MODEL

- **Multi Deteksi:** Model mampu mendeteksi banyak objek dalam satu gambar dengan akurasi spasial yang baik.
- **Adaptif terhadap variasi lingkungan:** Deteksi tetap berjalan baik meskipun citra diambil dalam kondisi berbeda, seperti cuaca mendung atau permukaan jalan basah.
- **Responsif terhadap kerusakan kecil:** Model mampu mendeteksi retakan kecil yang mungkin sulit terlihat secara kasat mata.

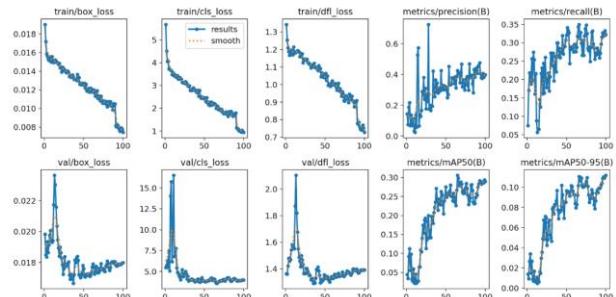
KETERBATASAN YANG DITEMUKAN

- Pada beberapa gambar dengan latar belakang kompleks atau pencahayaan

rendah, kemungkinan masih ada **objek yang tidak terdeteksi**, khususnya lubang jalan dengan warna yang menyatu dengan aspal.

- Terdapat sedikit **prediksi ganda** (*overlapping bounding box* untuk kerusakan yang sama), yang mungkin disebabkan oleh *noise* dalam anotasi atau area kerusakan yang sangat kompleks.

HASIL MODEL



Gambar 4. Grafik Train Loss, Validation Loss, Precision, Recall, dan mAP

Pemodelan deteksi jalan rusak menggunakan arsitektur YOLO menunjukkan hasil pelatihan dan validasi yang dapat dikatakan stabil dan mengalami perbaikan seiring bertambahnya *epoch*. Proses pelatihan dilakukan hingga 100 *epoch*, dan grafik menunjukkan evolusi dari berbagai komponen loss serta matrik evaluasi.

TRAIN LOSS

- *box_loss*, *cls_loss*, dan *dfl_loss* pada data pelatihan menunjukkan tren penurunan yang konsisten. Hal ini mengindikasikan bahwa model berhasil belajar secara efektif selama proses pelatihan.
- *box_loss* menurun dari sekitar 0.018 ke 0.008.
- *cls_loss* mengalami penurunan signifikan dari sekitar 5 ke 1.
- *dfl_loss* juga menurun secara bertahap dari 1.3 ke 0.75.

VALIDATION LOSS

- Meskipun terdapat fluktuasi pada awal *epoch* (terutama pada *cls_loss* dan *dfl_loss*), grafik menunjukkan stabilisasi setelah sekitar *epoch* ke-20.

- Penurunan val/cls_loss dari sekitar 15 ke 5, dan val/dfl_loss dari 2 ke sekitar 1.3, menunjukkan peningkatan generalisasi model terhadap data validasi.

PRECISION DAN RECALL

- Precision ($metrics/precision(B)$) meningkat secara bertahap, meskipun masih mengalami fluktuasi, dan stabil di sekitar 0.4 pada akhir pelatihan.
- Recall ($metrics/recall(B)$) menunjukkan peningkatan yang cukup signifikan dari 0.05 ke lebih dari 0.3, menunjukkan bahwa model semakin mampu mendeteksi lebih banyak objek jalan rusak dengan benar.

MEAN AVERAGE PRECISION (mAP)

- $metrics/mAP50(B)$ meningkat secara konsisten dari bawah 0.1 ke sekitar 0.27. Ini menandakan bahwa model semakin akurat dalam mendeteksi jalan rusak dengan ambang *threshold* IoU 50%.
- $metrics/mAP50-95(B)$ juga meningkat dari sekitar 0.01 ke lebih dari 0.1, menunjukkan performa model dalam berbagai tingkat kesulitan (IoU dari 0.5 hingga 0.95).

PERSAMAAN

$$\mathcal{L}_{total} = \lambda_{box} \cdot \mathcal{L}_{box} + \lambda_{cls} \cdot \mathcal{L}_{cls} + \lambda_{obj} \cdot \mathcal{L}_{obj} \quad (1)$$

- \mathcal{L}_{total} = Fungsi loss total yang digunakan untuk melatih model YOLO.
- \mathcal{L}_{box} = Fungsi loss untuk regresi bounding box.
- \mathcal{L}_{cls} = Fungsi loss untuk klasifikasi objek.
- \mathcal{L}_{obj} = Fungsi loss untuk prediksi keberadaan objek (objectness).
- λ_{box} = Koefisien penimbang untuk loss regresi bounding box
- λ_{cls} = Koefisien penimbang untuk loss klasifikasi objek.
- λ_{obj} = Koefisien penimbang untuk loss objectness

$$\mathcal{L}_{box} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v \quad (2)$$

- \mathcal{L}_{box} = Fungsi loss untuk regresi bounding box

- IoU = *Intersection over Union*, yaitu rasio antara area perpotongan dan gabungan dua kotak (prediksi dan ground truth).
- $\rho^2(\mathbf{b}, \mathbf{b}^{gt})$ = Kuadrat jarak Euclidean antara titik pusat kotak prediksi dan kotak ground truth.
- c = Panjang diagonal dari kotak terkecil yang mencakup kotak prediksi dan ground truth.
- v = Ukuran perbedaan aspek rasio antara kotak prediksi dan ground truth.
- α = koefisien yang digunakan untuk mengatur kontribusi v terhadap total loss.

$$IoU = \frac{A_{pred} \cap A_{gt}}{A_{pred} \cup A_{gt}} \quad (3)$$

- A_{pred} = Area dari kotak prediksi (*predicted bounding box*).
- A_{gt} = Area dari kotak *ground truth*.

$$\mathcal{L}_{cls} = - \sum_{i=1}^C [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (4)$$

- \mathcal{L}_{cls} = Fungsi loss untuk klasifikasi kelas objek
- C = jumlah total kelas dalam dataset
- y_i = label sebenarnya (*ground truth*) untuk kelas ke- i (bernilai 1 jika benar, 0 jika tidak)
- p_i = Panjang probabilitas hasil prediksi untuk kelas ke- i .

$$\mathcal{L}_{obj} = - [y_{obj} \log(p_{obj}) + (1 - y_{obj}) \log(1 - p_{obj})] \quad (5)$$

- \mathcal{L}_{obj} = Fungsi loss untuk memprediksi keberadaan objek (objectness).
- y_{obj} = label *ground truth* yang menunjukkan apakah ada objek dalam kotak prediksi (1 jika ada, 0 jika tidak).
- p_{obj} = Probabilitas prediksi bahwa terdapat objek dalam kotak tersebut

PENUTUP

SIMPULAN

Penelitian ini menunjukkan bahwa model YOLOv8 dapat diterapkan secara efektif untuk mendeteksi kerusakan jalan, khususnya jenis berlubang dan retak, melalui citra visual. Dengan memanfaatkan 200 gambar yang dianotasi secara manual dan dibagi dalam skema pelatihan yang proporsional, model mampu menghasilkan deteksi yang akurat dan stabil. Hasil ini mendukung tujuan

penelitian untuk menghadirkan solusi berbasis visi komputer yang dapat menggantikan inspeksi manual yang cenderung subjektif dan memerlukan banyak sumber daya. Temuan ini juga membuka peluang pengembangan lebih lanjut dalam sistem monitoring infrastruktur secara otomatis dan *real-time*.

SARAN

Berdasarkan hasil penelitian, terdapat beberapa saran untuk pengembangan lebih lanjut. Pertama, perlu penambahan jumlah dan variasi data guna meningkatkan generalisasi model, khususnya terhadap variasi pencahayaan, sudut pengambilan, dan jenis permukaan jalan. Kedua, sistem deteksi ini dapat dikembangkan menjadi aplikasi *real-time* melalui integrasi dengan kamera kendaraan atau perangkat bergerak lainnya. Penelitian lanjutan juga disarankan untuk menguji model pada skala data yang lebih besar dan beragam, serta mengevaluasi performa di perangkat komputasi terbatas seperti *edge devices*, agar sistem dapat diterapkan langsung di lapangan dengan efisiensi dan akurasi yang terjaga.

DAFTAR PUSTAKA

- Abdiansyah, F., Santoso, K., & Kamsyakawuni, A. (2021). Perbandingan Image RGB dan Grayscale pada Pengkodean Image dengan Algoritma 3D Playfair. PRISMA, Prosiding Seminar Nasional Matematika, 4, 524-533.
<https://journal.unnes.ac.id/sju/prisma/article/view/45012>
- Chau, M., & Reith, R. (2020). IDC - Smartphone Market Share - Vendor. Retrieved January 8, 2021, from Smartphone Market Share website: <https://www.idc.com/promo/smartphone-market-share/vendor>
- Chaki, J., & Dey, N. (2019). A beginner's guide to image preprocessing techniques. CRC Press.
- Erwi, S. G. V. K. (2024, 5). (H. Irsyad, Ed.). Implementasi Deteksi Objek Pada Jalan Rusak Menggunakan Metode YOLOv8, 3, 1-9. [10.58369/biit.v2i3.76](https://doi.org/10.58369/biit.v2i3.76)
- Hidayat, T., Firmansyah, R. F., Ilham, M., Yazid, M. N., & Rosyani, P. (2023). Deteksi pelanggaran sepeda motor pada marka jalan menggunakan algoritma YOLO dan mean average precision. Jurnal Riset Informatika dan Inovasi (JRIN), 1(2), 504-509.
- <https://doi.org/10.47970/siskom-kb.v8i1.401>
- Irfansyah, F. D., Kusuma, N. P., Renaldi, R. P., & Rosyani, P. (2024). Perancangan pendekripsi objek menggunakan metode YOLO dan OpenCV. Jurnal AI dan SPK: Jurnal Artificial Intelligence dan Sistem Penunjang Keputusan, 2(1), 43-47. https://jurnalmahasiswa.com/index.php/aidan_spk/article/view/1491
- Mahar, E. R., Sucipto, B. A., & Putri, R. R. M. (2025). Pengembangan sistem deteksi lubang pada jalan menggunakan algoritma YOLO berbasis ESP32-CAM. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 9(4), 1-12. <https://jptiik.ub.ac.id/index.php/jptiik/article/view/14713>
- Masitoh, S. (2018). Blended Learning Berwawasan Literasi Digital Suatu Upaya Meningkatkan Kualitas Pembelajaran dan Membangun Generasi Emas 2045. Proceedings of the ICECRS, 1(3), 13-34. <https://doi.org/10.21070/picecrs.v1i3.1377>
- Maszeri, M., Shohibuddin, M., et al. (2025). Deep learning dalam pendidikan dan artificial intelligence. Yayasan Putra Adi Dharma.
- Mishra, M. (2020, August 26). Convolutional neural networks, explained. Towards Data Science. Retrieved from <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939/>
- Padilla, R., Passos, W. L., Dias, T. L., Netto, S. L., & da Silva, E. A. (2021). A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics, 10(3), 279. <https://doi.org/10.3390/electronics10030279>
- Pramestya, R. H. (2018). Deteksi Dan Klasifikasi Kerusakan Jalan Aspal Menggunakan Metode Yolo Berbasis Citra Digital. Institut Teknologi Sepuluh Nopember Surabaya.
- Raup, A., Ridwan, W., Khoeriyah, Y., Supiana, & Zaqiah, Q. Y. (2022). Deep learning dan penerapannya dalam pembelajaran. JIIP (Jurnal Ilmiah Ilmu Pendidikan), 5(9), 3258-3267. <http://jiip.stkipyapisdompu.ac.id/index.php/JIIP/article/view/5777>
- Tilasefana, R. A., & Putra, R. E. (2023). Penerapan metode deep learning menggunakan algoritma CNN dengan arsitektur VGG Net untuk pengenalan cuaca. JINACS (Journal of Informatics and Computer Science), 5(1), 48-57. <https://doi.org/10.26740/jinacs.v5n01.p48-57>
- Utomo, A. N. (2020, 12). Analisa Data Ekstraksi Ciri Citra Momen Histogram Dan Perbandingan Model Algoritma Klasifikasi Naive Bayes, Nearest Neighbor, Support Vector Machine, Dan Decision Tree Pada Studi Kasus Citra Jalan Aspal Rusak Dan Jalan Aspal Tidak Rusak,