

Sistem Cerdas Deteksi Kerusakan Jalan dengan Perbandingan Algoritma YOLOV8 dan SSD MobileNet

Jupriola Jeskris Soleman Tualaka^{1*}, Vihi Atina², Anisatul Farida³

¹Fakultas Ilmu Komputer
Universitas Duta Bangsa

¹*220103062@mhs.udb.ac.id (penulis
korespondensi)

²Fakultas Ilmu Komputer
Universitas Duta Bangsa

²vihi_atina@udb.ac.id

³Fakultas Ilmu Komputer
Universitas Duta Bangsa

³anisatul_farida@udb.ac.id

Abstrak— Infrastruktur jalan raya yang aman dan nyaman merupakan fasilitas esensial bagi masyarakat, namun pengawasan kondisi jalan secara manual memiliki keterbatasan yang signifikan. Penelitian ini bertujuan untuk mengembangkan sistem cerdas pendeteksi kerusakan jalan raya guna mengenali berbagai jenis kerusakan seperti retak buaya, pelepasan butir, dan lubang. Sebagai solusi atas permasalahan tersebut, penelitian ini membandingkan kinerja dua algoritma *deep learning* berbasis *Convolutional Neural Network* (CNN) tipe *one-stage detector*, yaitu YOLOv8 dan SSD MobileNet. Metodologi yang digunakan adalah *Cross-Industry Standard Process for Data Mining* (CRISP-DM). Dataset yang digunakan sangat terbatas, yakni 45 citra kerusakan jalan yang diekstraksi dari *Google Maps* dan tangkapan kamera langsung, kemudian dikelola serta diaugmentasi menggunakan platform Roboflow. Hasil pengujian menunjukkan bahwa algoritma YOLOv8 memiliki tingkat responsivitas yang lebih tinggi dengan nilai *mean Average Precision* (mAP) pada IoU 0.50 berhasil mencapai angka 72 persen. Namun, metrik YOLOv8 menunjukkan volatilitas dan indikasi *overfitting* akibat menghafal dataset latih yang terlalu sempit. Di sisi lain, model SSD MobileNet mengalami kegagalan konvergensi secara mutlak dengan nilai mAP yang terus stagnan di angka 30 persen selama 100 *epoch*. Kegagalan ini disebabkan oleh ketidakmampuan algoritma SSD dalam menangani gangguan visual atau *noise* pada kondisi data yang sangat minim. Dapat disimpulkan bahwa kedua model saat ini belum siap untuk tahap implementasi produksi.

Kata kunci— Deteksi Kerusakan; Jalan Raya; *Deep Learning*; YOLOv8; SSD MobileNet

Abstract— Safe and comfortable highway infrastructure is an essential amenity for the public, but manual monitoring of road conditions has significant limitations. This study aims to develop an intelligent highway damage detection system to identify various types of damage, such as alligator cracking, aggregate loss, and potholes. To address this issue, this study compares the performance of two one-stage detector-type Convolutional Neural Network (CNN)-based deep learning algorithms: YOLOv8 and SSD MobileNet. The methodology used is the Cross-Industry Standard Process for Data Mining (CRISP-DM). The dataset used was very limited, consisting of 45 images of road damage extracted from Google Maps and direct camera captures, which were then processed and augmented using the Roboflow platform. Test results show that the YOLOv8 algorithm has a higher level of responsiveness, with a mean Average Precision (mAP) at an IoU of 0.50 reaching 72 percent. However, YOLOv8's metrics exhibit volatility and signs of overfitting due to memorizing a training dataset that was too narrow. On the other hand, the SSD MobileNet model failed to converge entirely, with its mAP value remaining stagnant at 30 percent for 100 epochs. This failure was caused by the SSD algorithm's inability to handle visual noise under conditions of extremely sparse data. It can be concluded that both model is neither model is currently ready for the production implementation phase.

Keywords— Damage Detection; Highways; Deep Learning; YOLOv8; SSD MobileNet

I. PENDAHULUAN

Jalan raya merupakan fasilitas yang penting bagi masyarakat agar dapat mencapai suatu tujuan yang diinginkannya, untuk itu masyarakat membutuhkan jalan raya yang aman dan nyaman bagi penggunaanya, serta diharapkan dapat meningkatkan perindustrian dan perekonomian masyarakat tersebut[1]. Infrastruktur jalan perkotaan sudah seharusnya tidak berlubang dan memenuhi standar perkerasan seperti ketebalan jalan yang cukup, memiliki perkerasan kaku agar dapat memikul beban yang bekerja dan memiliki

saluran permukaan mampu mengalirkan air dengan baik [2].

Dalam konteks penelitian ini, pemanfaatan citra digital sebagai dasar pendeteksian kerusakan jalan menjadi semakin relevan mengingat jumlah ruas jalan terus bertambah, sementara kemampuan pengawasan manual masih sangat terbatas. Penggunaan teknologi mampu menjembatani kesenjangan tersebut dengan menyediakan metode yang lebih terstandarisasi, cepat, dan hemat biaya.

Proses pencatatan perbaikan jalan rusak perlu ditingkatkan, salah satu caranya yaitu dengan mengenali ciri dan pola dari kerusakan jalan

tersebut. Pencatatan kondisi jalan rusak dapat menggunakan bantuan teknologi untuk mengklasifikasi citra jalan rusak berdasarkan ciri dan polanya, dengan menggunakan deep learning yang merupakan salah satu turunan dari machine learning [3]. *Machine learning* dan *deep learning* telah secara signifikan mempermudah serta mengurangi kompleksitas dan biaya terkait dengan hal tersebut. Para peneliti telah mengusulkan berbagai pendekatan untuk identifikasi lubang jalan [4]. Salah satu nya adalah menggunakan CNN atau *Convolutional Neural Networks*. Teknologi CNN juga memungkinkan deteksi berbagai jenis kerusakan secara lebih spesifik, seperti retak buaya, lubang, pelepasan butir, hingga deformasi pada permukaan jalan. Dengan kemampuan ekstraksi ciri yang lebih kuat, CNN merupakan model jenis deep learning yang terinspirasi oleh bagaimana manusia memahami dunia visual, dan model ini mampu digunakan untuk menganalisis data gambar [5]. Model CNN dapat mengenali pola visual yang mungkin diidentifikasi oleh pengamat manusia, terutama ketika kerusakan masih dalam tahap awal. Identifikasi dini (early detection) sangat penting karena dapat membantu pemerintah mencegah kerusakan yang lebih parah dan mengurangi biaya perbaikan yang berlipat ganda jika jalan sudah dalam kondisi rusak berat. Pendekatan ini tidak hanya memberikan efisiensi pemantauan kondisi jalan, namun juga menjadi langkah strategis untuk mendukung pengambilan keputusan terkait prioritas perbaikan dan pengelolaan infrastruktur.

Dalam pengembangan sistem cerdas untuk deteksi objek, algoritma deep learning berbasis *Convolutional Neural Network* tipe one-stage detector seperti YOLO dan SSD sering diimplementasikan karena strukturnya yang praktis dan kemampuannya untuk menangani skenario aplikasi real-time. Berdasarkan penelitian terdahulu, algoritma YOLO memiliki keunggulan yang signifikan dalam hal kecepatan inferensi, akurasi, dan stabilitas hasil deteksi. YOLO mampu melakukan prediksi langsung secara komprehensif tanpa bergantung pada anchor boxes, sehingga sangat optimal untuk memangkas beban komputasi secara real-time

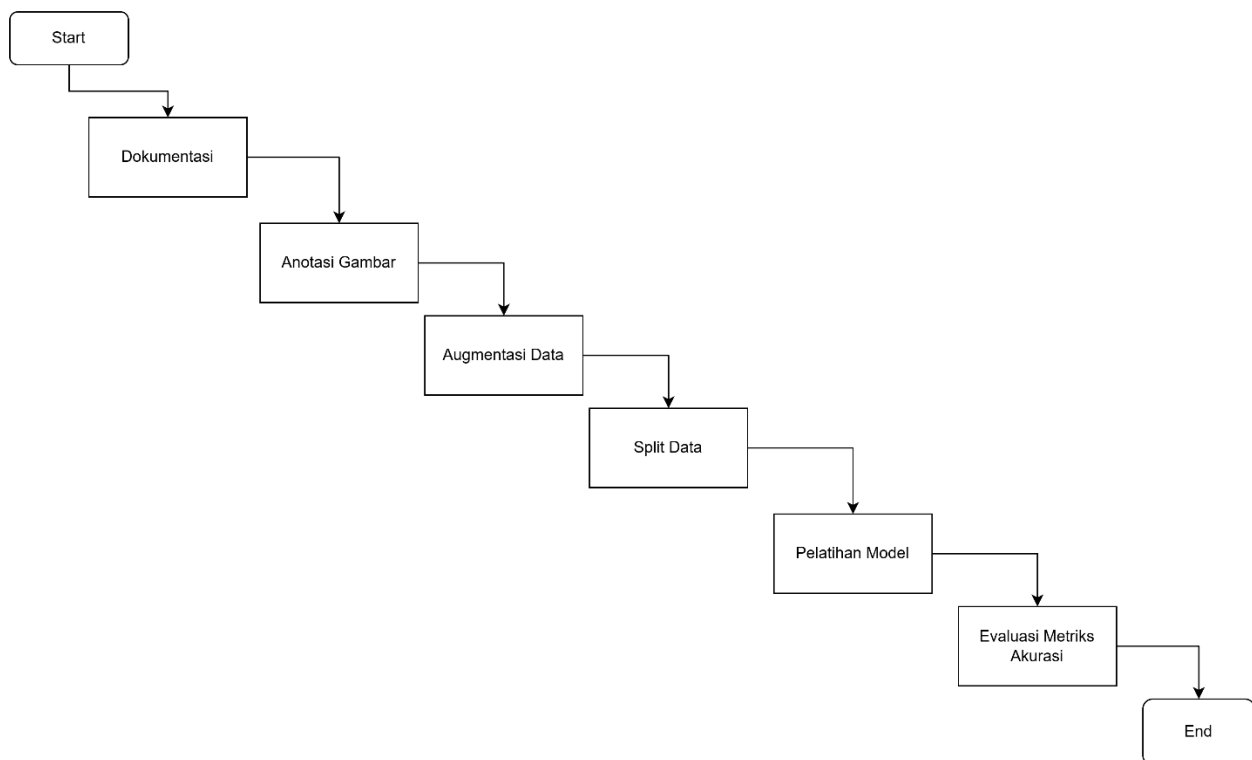
[6]. Namun, metode YOLO memiliki kelemahan pada tingginya konsumsi RAM dan besarnya kapasitas penyimpanan yang dibutuhkan, sehingga rentan membebani perangkat dengan sumber daya terbatas [6]. Sebagai alternatif komparatif, metode SSD yang dipadukan dengan arsitektur *MobileNet* menawarkan kelebihan berupa efisiensi memori dan ruang penyimpanan yang jauh lebih ringan berkat operasi komputasi *depthwise separable convolutions* [6];[7]. Meskipun sangat efisien untuk perangkat mobile atau perangkat dengan spesifikasi rendah, metode SSD memiliki kelemahan pada waktu pemrosesan yang sedikit lebih lambat dibandingkan YOLO akibat penggunaan anchor boxes [6]. Selain itu, kinerja SSD terbukti lebih sensitif, di mana nilai akurasi dan recall cenderung menurun dan bervariasi ketika dihadapkan pada perubahan kondisi lingkungan, seperti pencahayaan yang berbeda atau kondisi objek yang ramai [8].

II. METODOLOGI PENELITIAN

Metode penelitian yang digunakan oleh penulis dalam melaksanakan penelitian ini dijadikan pedoman untuk memperoleh hasil yang sesuai dengan tujuan yang telah ditentukan sebelumnya [9]. Diagram alir penelitian dapat dilihat pada gambar 1 dan arsitektur model pada gambar 3.

A. Diagram Alir Penelitian

Tahapan pengumpulan data dilakukan melalui kombinasi dua pendekatan, yakni ekstraksi citra dari Google Maps dan pengambilan gambar secara langsung di lapangan menggunakan kamera ponsel. Pendekatan ini mengumpulkan total 45 data citra kerusakan jalan yang diklasifikasikan ke dalam dua jenis kelas yaitu pelepasan butir (*raveling*), dan lubang (*pothole*). Penggunaan citra dari Google Maps berfungsi



Gambar 1. Kerangka / Alur Penelitian

sebagai basis data awal untuk mengenali pola dasar kerusakan, sementara pengambilan citra langsung bertujuan untuk memperkaya variasi kondisi pencahayaan serta karakteristik jalan lokal. Seluruh dataset yang terkumpul kemudian dikelola menggunakan platform Roboflow untuk tahapan seleksi, anotasi, dan pra-pemrosesan (preprocessing) guna memastikan kualitas citra memenuhi kebutuhan pelatihan model CNN).

B. Tahapan Penelitian



Gambar 2. Tahapan CRISP-DM

Sumber: [10]

Metode pengembangan sistem yang digunakan pada penelitian ini adalah CRISP-DM (Cross-Industry Standard Process for Data Mining). Metode ini dipilih karena sesuai untuk penelitian berbasis kecerdasan buatan yang memerlukan proses iteratif dalam pengolahan data dan pembangunan model. Metode ini terdiri dari enam tahapan: business understanding, data understanding, data preparation, modeling, evaluation, deployment [11].

1. Business Understanding
Untuk memahami kebutuhan dan tujuan deteksi kerusakan jalan
2. Data Understanding
Untuk mempelajari karakteristik hasil citra maupun data citra kerusakan jalan yang diperoleh secara terbatas dari sumber terbuka maupun dokumentasi
3. Data Preparation
Mencakup seleksi data, labelling sederhana, dan augmentasi sebagai langkah awal peningkatan kualitas data
4. Modeling
Melatih dua arsitektur model berbeda yaitu YOLOv8 dan SSD *MobileNet*, menggunakan dataset yang sama untuk mendapatkan model hasil penelitian dari masing – masing algoritma.
5. Evaluation
Mengukur performa model berdasarkan akurasi dan metrik evaluasi dasar
6. Deployment
Tahap rencana implementasi hasil model ke *prototype* aplikasi web pendeteksi kerusakan jalan

C. Teknik Augmentasi Data dan Pra-pemrosesan

Untuk merekayasa variasi lingkungan visual dan mencegah *overfitting* akibat keterbatasan data, dilakukan pipa (*pipeline*) augmentasi data pada platform *Roboflow*. Parameter augmentasi diatur secara ketat agar tidak menghilangkan esensi fitur spasial dari kerusakan jalan raya. Detail parameter operasi augmentasi ditunjukkan pada Tabel 1.

Tabel 1. Tabel Augmentasi

Jenis Augmentasi	Parameter Spesifik	Tujuan Teknis
<i>Horizontal Flip</i>	Sumbu X (Kiri-Kanan)	Mensimulasikan arah laju kendaraan dari jalur berlawanan.
<i>Rotation</i>	Antara -15° hingga +15°	Mengantisipasi sudut kemiringan (<i>tilt</i>) kamera ponsel atau Google Maps.

<i>Bounding Box Blur</i>	Maksimum 2.5 piksel	Mensimulasikan efek <i>motion blur</i> akibat kendaraan bergerak cepat.
<i>Gaussian Noise</i>	0.5% dari total piksel	Mensimulasikan interferensi sensor kamera pada kondisi cuaca redup atau lensa kotor.

D. Formula Matematis Metrik Evaluasi

Untuk mengukur dan mengevaluasi kinerja arsitektur deep learning YOLOv8 dan SSD *MobileNet* secara objektif, penelitian ini menggunakan metrik evaluasi standar yang berlaku pada tugas deteksi objek (*object detection*). Metrik tersebut meliputi Precision, Recall, Average Precision (AP), dan Mean Average Precision (mAP).

a. Precision

Precision atau tingkat presisi mengukur kemampuan model dalam memberikan prediksi yang akurat. Metrik ini menunjukkan persentase objek yang benar-benar merupakan kerusakan jalan dari keseluruhan total objek yang diprediksi oleh model. Nilai *Precision* yang tinggi meminimalkan terjadinya deteksi palsu [12]. Formulasi matematis *Precision* ditunjukkan pada Persamaan (1):

$$Precision = \frac{TP}{TP+FP} \quad (1) \quad [13]$$

b. Recall

Recall atau sensitivitas mengukur keberhasilan model dalam menemukan kembali seluruh objek kerusakan jalan yang tersebar di dalam dataset. Metrik ini menunjukkan persentase objek kerusakan jalan yang berhasil dideteksi oleh model dari total keseluruhan objek kerusakan yang seharusnya ditemukan. Nilai *Recall* yang tinggi meminimalkan adanya kerusakan jalan yang terlewat dari pengawasan sistem [14]. Formulasi matematis *Recall* ditunjukkan pada Persamaan (2):

$$Recall = \frac{TP}{TP+FN} \quad (2) \quad [13]$$

c. Average Precision (AP) dan Mean Average Precision (mAP)

Karena nilai *Precision* dan *Recall* saling bertolak belakang (peningkatan salah satu metrik sering kali menurunkan metrik lainnya), performa model deteksi objek tidak dapat dinilai hanya dari satu sisi. Oleh karena itu, digunakan kurva *Precision-Recall (PR-Curve)*. *Average Precision (AP)* adalah nilai rata-rata dari *Precision* yang dihitung pada setiap tingkat *Recall* yang berbeda di sepanjang kurva tersebut. Secara matematis, AP dihitung dengan mencari luas daerah di bawah kurva *Precision-Recall (AUC-PR)* melalui integrasi numerik, seperti yang dirumuskan pada Persamaan

$$AP = \sum_{i=1}^n (R_i - R_{i-1}) P_i \quad (3) \quad [15]$$

Di mana $p(r)$ adalah fungsi *Precision* sebagai fungsi dari *Recall* (r). Untuk mendeteksi multi-kelas (retak buaya, pelepasan butir, dan lubang), digunakan Mean Average Precision (mAP). Nilai mAP diperoleh dengan menghitung rata-rata nilai AP dari seluruh kelas objek (N) yang ada pada penelitian, sesuai dengan Persamaan (4):

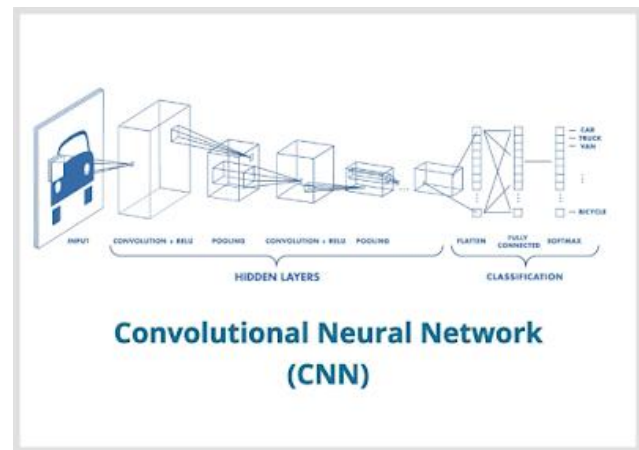
$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4) \quad [15]$$

Pada penelitian ini, evaluasi difokuskan pada nilai $mAP@0.50$, yang berarti nilai mAP dihitung pada saat ambang batas IoU berada tepat di angka 0.50. Metrik inilah yang menjadi acuan utama untuk menentukan tingkat responsivitas, stabilitas, dan kesiapan operasional dari model YOLOv8 dan SSD *MobileNet* sebelum masuk ke tahap implementasi [16].

E. Sistem Cerdas

Sistem cerdas yang mampu menerapkan pengetahuan dan inferensi prosedur untuk mengatasi permasalahan yang cukup rumit yang memerlukan keahlian seorang pakar untuk memecahkan solusi. Sistem yang mampu mengelola kemampuan manusia yang direkam oleh komputer untuk menyelesaikan berbagai permasalahan dengan membutuhkan keahlian manusia [17]. Dalam sistem cerdas, terdapat metode-metode yang membantu dalam penelitian seperti deep learning termasuk pada pendeteksian kerusakan jalan.

F. Metode Convolutional Neural Network (CNN)



Gambar 3. Arsitektur model CNN

Sumber : [18]

Convolutional Neural Network (CNN) merupakan salah satu algoritma yang populer dalam *deep learning* dan dapat digunakan untuk Computer Vision atau memiliki kemampuan mengolah informasi citra. CNN mempunyai keunggulan yang tak tertandingi dalam mengekstraksi fitur gambar. Model CNN banyak digunakan untuk tugas *image processing* karena memiliki tingkat akurasi yang tinggi dan lebih baik dalam pengenalan citra. Ekstraksi tekstur dengan CNN adalah metode yang efektif untuk mengenali permukaan jalan yang mulus, retak, atau berlubang. CNN, yang meniru sistem pengenalan citra pada otak manusia, terbukti signifikan dalam pengenalan citra [19]. Setelah melalui tahapan ekstraksi fitur pada data gambar, proses berikutnya di dalam arsitektur

CNN adalah melakukan pelatihan model. Fase pelatihan ini secara umum dibagi menjadi dua tingkatan utama, yaitu *feature learning* dan *classification*. Di dalam tahapan *feature learning*, dimensi spasial dari citra masukan akan direduksi secara bertahap lewat operasi pada *convolutional layer* serta *max pooling layer*. Untuk menentukan dimensi spasial dari feature map yang dihasilkan oleh convolutional layer, digunakan formulasi matematis sebagai Persamaan (5):

$$n_{(w,h)} = \left\lceil \frac{n_{in} + 2p - k}{s} \right\rceil + 1 \quad (5) \quad [13]$$

Keterangan:

- $n_{(w,h)}$ = Dimensi width atau height dari feature map yang dihasilkan.
- k = Dimensi dari kernel atau filter yang diterapkan.
- s = Nilai langkah pergeseran filter.
- p = Jumlah baris/kolom nilai kosong yang ditambahkan pada tepi citra.
- n_{in} = Dimensi spasial dari citra masukan sebelum konvolusi.

Setelah ukuran spasial dari citra diperoleh dari convolutional layer, tahapan *feature learning* berikutnya di dalam CNN dilanjutkan pada komponen *max pooling layer*. Fungsi dari lapisan ini adalah untuk mereduksi ukuran data dengan mengambil nilai maksimum pada area tertentu. Penghitungan dimensi spasial setelah melewati *max pooling layer* diformulasikan melalui rumus Persamaan (6):

$$n_{(w,h)} = \left\lceil \frac{(n_{(w,h)} - 1 - f)}{s} \right\rceil + 1 \quad (6) \quad [13]$$

G. Citra Digital

Secara sederhana, citra merupakan gambar pada bidang dua dimensi (2D). Citra dapat memberikan informasi dimensi yang ada pada citra tersebut berkat adanya cahaya ketika menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya pada mata manusia, kamera, scanner, dan sebagainya, sehingga bayangan pada objek tersebut merekam [20].

Citra digital digunakan pada penelitian untuk membantu visual pada sebuah gambar seperti pada kerusakan jalan. Deteksi kerusakan jalan melalui tangkapan kamera dapat dilakukan dengan mendeteksi adanya garis tepi yang tebal [21].

H. Deteksi Kerusakan Jalan

Deteksi kerusakan jalan bertujuan untuk mengetahui seberapa parah sebuah jalan mengalami kerusakan. Kerusakan jalan inilah yang menyebabkan banyaknya kecelakaan pada lalu lintas.

Adanya air yang menggenangi jalan saat hujan adalah penyebab utama kerusakan jalan karena air dapat melonggarkan ikatan aspal dan agregat. Ketika ikatan aspal dan agregat longgar, kendaraan yang lewat akan memberi beban, menyebabkan retak atau kerusakan jalan lainnya [22].

I. YOLOv8

YOLOv8 adalah YOLO terbaru untuk sistem deteksi objek. Algoritma YOLOv8 dikembangkan oleh Alexy Bochkovskiy dengan menggunakan CSPDarknet53 sebagai arsitektur backbone yang memiliki efisiensi komputasi yang lebih baik [23].

J. SSD-MobileNet

SSD merupakan algoritma deteksi objek yang menggunakan pendekatan one-stage, SSD memanfaatkan multiple feature dari layer-layer CNN dengan resolusi berbeda untuk mendeteksi objek dalam berbagai skala [8]. Untuk meminimalkan ukuran dan kompleksitas model, arsitektur *MobileNet* menggunakan konvolusi yang dipisahkan berdasarkan kedalaman (konvolusi kedalaman dan konvolusi titik). Elemen utama dari model ini adalah pengali lebar dan pengali resolusi. Keduanya disesuaikan selama proses pelatihan untuk memenuhi persyaratan kecepatan pemrosesan dan ukuran yang ringkas [7].

III. HASIL DAN PEMBAHASAN

Penelitian ini mengevaluasi kinerja sistem cerdas pendeteksi kerusakan jalan raya dengan

membandingkan algoritma YOLOv8 dan SSD *MobileNet*. Hasil eksperimen pada lingkungan komputasi Google Colab dianalisis secara terstruktur menggunakan fase-fase dalam kerangka kerja CRISP-DM. Pada setiap fasenya, temuan yang didapat akan dievaluasi guna menjawab hipotesis awal penelitian dan disandingkan dengan hasil kajian literatur dari penelitian-penelitian sejenis sebelumnya.

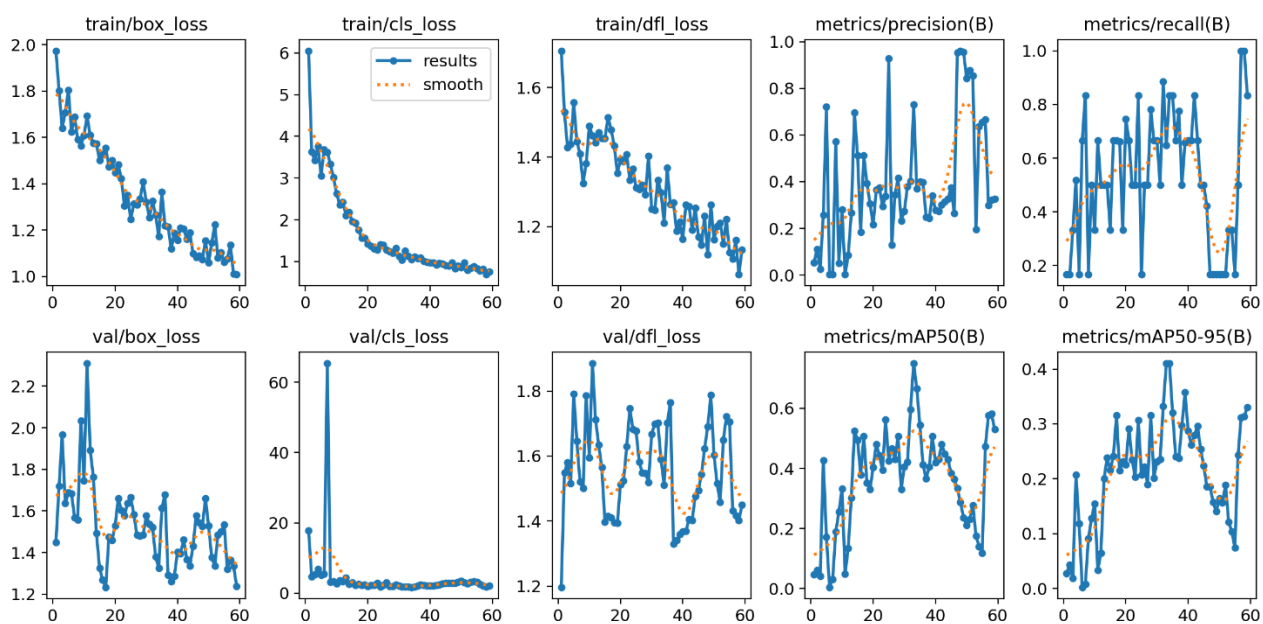
A. Business Understanding

Hipotesis awal dalam penelitian ini mengasumsikan bahwa penggunaan arsitektur algoritma *one-stage detector* mampu mendeteksi dan mengklasifikasikan objek kerusakan jalan (seperti lubang dan pengelupasan) dengan tingkat akurasi dan kecepatan yang seimbang.

ravelling yang memiliki stabilitas prediksi tinggi. Namun, berdasarkan hasil pengujian komparatif dengan kondisi data yang sangat terbatas (40 citra), luaran dari kedua model belum sepenuhnya memenuhi kriteria operasional. YOLOv8 menunjukkan potensi responsivitas yang tinggi, namun metriknya memperlihatkan volatilitas. Sementara itu, SSD *MobileNet* gagal mencapai standar akurasi minimum. Hasil ini membuktikan bahwa efektivitas arsitektur *Deep Learning* tidak hanya bergantung pada algoritma, tetapi terikat pada volume ketersediaan data.

B. Data Understanding

Kondisi dataset latihan memberikan dampak yang sangat signifikan terhadap metrik akhir pemodelan. Dalam penelitian ini, model dilatih menggunakan dataset yang memiliki keterbatasan dari segi kuantitas. Karakteristik data yang terbatas ini menjadi tantangan besar,



Gambar 4 Grafik Pelatihan, Validasi, dan Evaluasi Algoritma YOLOv8

Asumsi ini sejalan dengan karakteristik dasar algoritma YOLO yang menerapkan pendekatan *grid-based prediction* dan SSD yang mengandalkan multi-feature maps dari beberapa tingkatan layer jaringan konvolusi [8].

Tujuan utama eksperimen ini adalah menghasilkan model pendeteksi pothole dan

khususnya bagi arsitektur tertentu seperti SSD *MobileNet*. Berdasarkan kajian literatur, performa algoritma SSD terbukti sangat sensitif terhadap variasi serta perubahan kondisi lingkungan pada data masukan. Pada kondisi lingkungan yang tidak ideal, nilai sensitivitas (recall) dari SSD dapat berfluktuasi secara drastis dalam rentang 72% hingga 89% [8]. Keterbatasan jumlah data asli dan kompleksitas tekstur kerumitan aspal bawaan menyebabkan

batas-batas spasial objek menjadi kurang tegas sejak awal.

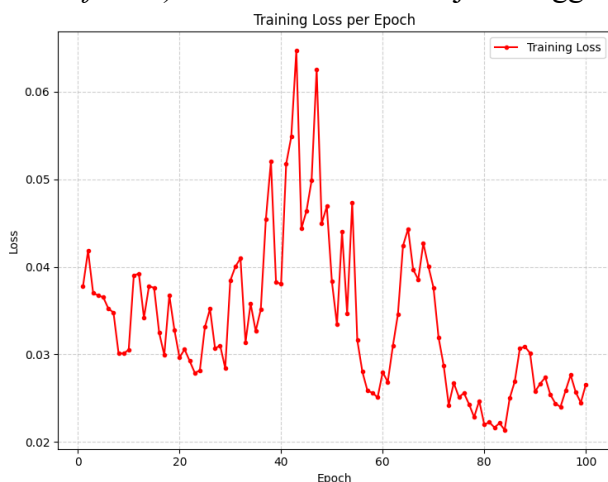
C. Data Preparation

Kondisi set data latih memberikan dampak yang sangat signifikan terhadap metrik akhir pemodelan. Model dilatih menggunakan dataset terbatas yang kemudian diproses menggunakan fitur augmentasi berupa *Flip Horizontal*, Rotasi - 15 hingga 15 derajat, *Blur* 2.5 piksel, dan *Noise* 0.5 persen.

D. Modeling

Perbedaan drastis pada grafik kedua model ini dapat dijelaskan melalui perbedaan fundamental arsitektur mereka.

Dinamika Loss YOLOv8: Berdasarkan metrik pelatihan (lihat Gambar 3), pergerakan kerugian latih (*train box loss*, *train cls loss*, dan *train dfl loss*) YOLOv8 menurun tajam hingga



Gambar 5 Grafik Training Loss dan Metrik Evaluasi SSD MobileNet

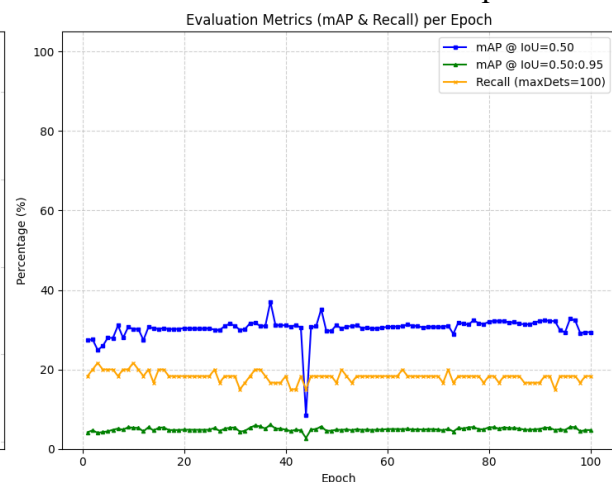
mencapai nilai mendekati 1.0 pada epoch ke-60. Namun, kurva validasi bergerak sangat fluktuatif dengan lonjakan *error* ekstrem pada epoch ke-10 hingga ke-30. Ketidakseimbangan ini membuktikan terjadinya *overfitting*. YOLOv8 terlalu sensitif dan menghafal dataset latih yang sempit sehingga kehilangan kemampuan generalisasinya.

Dinamika Loss SSD MobileNet: Sebaliknya, grafik fungsi kerugian SSD MobileNet (lihat Gambar 4) menunjukkan kegagalan konvergensi mutlak. Nilai kerugian melompat tajam menembus angka 0.065 pada pertengahan *epoch*.

Kegagalan SSD ini dapat dipahami mengingat SSD membutuhkan keragaman fitur citra yang tinggi untuk mengoptimalkan mekanisme *multi-feature maps* dalam mendeteksi objek dalam berbagai skala [8]. Karena arsitektur *MobileNet* tidak memiliki sampel yang cukup untuk mengekstrak fitur *ravelling*, bobot jaringan terus diperbarui secara acak dan memicu kemacetan belajar.

E. Evaluation

Evaluasi komparatif metrik akurasi menegaskan perbedaan kapabilitas struktural dari kedua arsitektur saat menangani objek ireguler. Algoritma YOLOv8 mencatatkan performa presisi yang sangat dinamis, di mana nilai mAP pada IoU 0.50 berhasil merangkak naik menyentuh 72 persen. Superioritas kemampuan adaptasi algoritma YOLO pada lingkungan terbatas ini relevan dengan temuan studi identifikasi biometrik komparatif yang



membuktikan bahwa keluarga YOLO (seperti YOLOv5) secara konsisten mampu mengungguli MobileNet-SSD (V1 maupun V2) baik dari segi akurasi maupun efisiensi ukuran arsitektur [7].

Selain melakukan evaluasi berdasarkan angka-angka metrik kuantitatif, analisis kualitatif juga dilakukan dengan menguji berkas bobot (*weights*) final model YOLOv8 untuk melakukan inferensi langsung pada sampel citra pengujian kondisi jalan lokal. Gambar 5 menampilkan hasil representasi visual prediksi kotak pembatas (*bounding box*) yang dihasilkan oleh algoritma.



Gambar 6. Hasil Deteksi YOLOv8

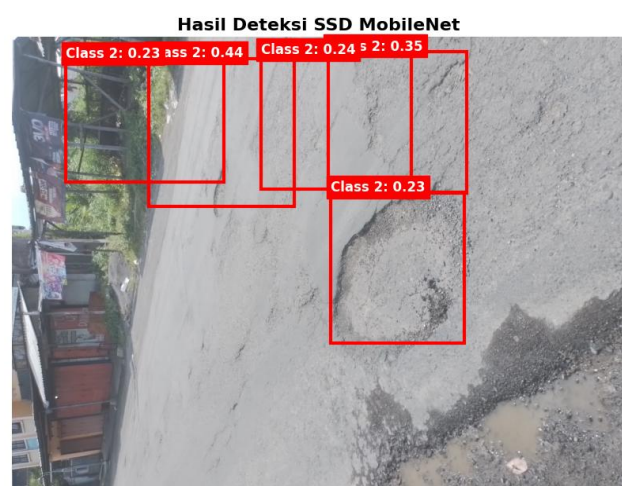
Berdasarkan hasil uji visual pada antarmuka deteksi di atas, model YOLOv8 terbukti mampu melokalisasi target secara mandiri. Fungsionalitas visual dari model berhasil mendeteksi beberapa area lubang jalan (pothole) dengan tingkat kepercayaan (confidence score) yang bersifat moderat, yakni berada pada rentang 0.38 hingga 0.66. Angka ini sejalan dengan keterbatasan jumlah *epoch* dan minimnya variasi dataset awal yang digunakan pada saat pelatihan.

Namun, pengamatan mendalam terhadap representasi visual ini juga menyingkap dua anomali teknis yang mendasari volatilitas metrik evaluasi model. **Pertama**, terjadi penumpukan kotak pembatas (*overlapping bounding boxes*) pada satu objek lubang makro di area tengah citra (dengan skor 0.66, 0.66, dan 0.62). Penumpukan ini mengindikasikan bahwa model berhasil mendeteksi fitur dari beberapa lapisan *grid* yang berbeda, namun memerlukan penyesuaian kalibrasi lanjutan pada parameter ambang batas *Non-Maximum Suppression* (NMS) saat implementasi UI agar sistem hanya memfilter dan menampilkan satu kotak yang paling presisi. **Kedua**, model mengalami kegagalan deteksi (*missed detection* atau *False Negative*) pada area kerusakan jalan yang tergenang air (terlihat pada sudut kiri bawah gambar). Genangan air secara optik mendistorsi pantulan cahaya dan mengubah

tekstur asli dari aspal yang rusak. Karena model YOLOv8 hanya dilatih menggunakan 45 citra jalan kering dan hanya direkayasa menggunakan blur serta noise pada tahap augmentasi, model tidak memiliki referensi memori untuk mengenali bahwa genangan tersebut adalah sebuah lubang.

Fenomena ini membuktikan hipotesis penelitian bahwa meskipun arsitektur anchor-free milik YOLOv8 sangat responsif dalam menemukan objek secara umum, ketahanannya (robustness) sangat rentan ketika dihadapkan pada deviasi lingkungan visual riil (seperti air atau pencahayaan ekstrem) yang tidak terwakili secara komprehensif di dalam data training.

Di sisi lain, SSD MobileNet mencatatkan stagnasi mAP di angka 30 persen selama 100 epoch. Kegagalan SSD dalam penelitian ini memberikan kontras yang tajam jika disandingkan dengan penelitian literatur lainnya. Sebuah pengujian deteksi kerusakan jalan secara spesifik pada perangkat mobile membuktikan bahwa apabila dilatih dengan data yang optimal, SSD-MobileNet justru mampu mencatatkan tingkat akurasi keseluruhan sebesar 91,1%, mengalahkan YOLOv8 yang berada di angka 86,4% [6]. Hal ini menegaskan kesimpulan bahwa kegagalan metrik SSD pada eksperimen saat ini murni akibat dari ketidakcukupan jumlah data latih, bukan karena kecacatan algoritma.



Gambar 7. Hasil deteksi SSD MobileNet

Berdasarkan pengujian visual pada gambar jalan berlubang (Gambar 6), terdapat perbedaan

performa yang signifikan antara arsitektur YOLO dan SSD *MobileNet*. Pada hasil deteksi menggunakan SSD *MobileNet*, model mampu mengenali keberadaan objek (Class 2), namun menghasilkan tingkat kepercayaan (confidence score) yang tergolong rendah, yakni berada pada rentang 0.23 hingga 0.44. Selain itu, lokalisasi bounding box pada SSD *MobileNet* kurang presisi, ditandai dengan adanya tumpang tindih (*overlapping*) kotak deteksi dan indikasi *false positive* pada area jalan yang tidak rusak parah.

F. Deployment

Berdasarkan hasil pemodelan, disimpulkan bahwa file bobot (weights) kedua model belum siap untuk tahap implementasi produksi. File bobot SSD *MobileNet* gagal mendeteksi objek, sementara YOLOv8 menyimpan risiko overfitting yang berpotensi menghasilkan deteksi palsu (*false positive*) pada antarmuka visual.

Selain aspek akurasi, tinjauan literatur membuktikan bahwa tingkat kesiapan deployment juga ditentukan oleh manajemen sumber daya. Implementasi arsitektur YOLOv8 diketahui menuntut alokasi memori yang cukup tinggi, memakan kapasitas penyimpanan hingga 136 MB dan beban RAM sebesar 0,41 GB [6]. Sebaliknya, SSD-*MobileNet* menawarkan efisiensi absolut yang dikhususkan untuk perangkat terbatas, dengan kebutuhan penyimpanan hanya sebesar 18,8 MB dan konsumsi RAM 0,14 GB[6] .

Oleh karena itu, meskipun YOLOv8 memberikan hasil metrik yang lebih responsif pada kondisi krisis data, rancangan pengembangan sistem pada iterasi selanjutnya wajib memprioritaskan perbaikan data untuk SSD-*MobileNet*. Dengan menambah kuantitas sampel aspal rusak secara masif, SSD-*MobileNet* diyakini mampu mencapai konvergensi dan tingkat akurasi optimal, menjadikannya pilihan algoritma yang paling efisien untuk diintegrasikan pada sistem deteksi fungsional.

IV. KESIMPULAN

Penelitian ini telah berhasil melakukan evaluasi dan perbandingan terhadap kinerja sistem cerdas pendeteksi kerusakan jalan raya

menggunakan algoritma YOLOv8 dan SSD *MobileNet*. Berdasarkan hasil pengujian yang dilakukan melalui kerangka kerja CRISP-DM, ditemukan jawaban atas permasalahan penelitian bahwa arsitektur YOLOv8 menunjukkan tingkat responsivitas dan adaptasi yang lebih superior pada kondisi krisis data dibandingkan SSD *MobileNet*. Algoritma YOLOv8 berhasil mencatatkan presisi dinamis dengan nilai *mean Average Precision* (mAP) pada IoU 0.50 yang mencapai 72 persen. Meskipun demikian, hasil akhir menunjukkan bahwa bobot (*weights*) dari kedua model tersebut belum memenuhi standar kesiapan untuk tahap implementasi produksi atau *deployment* ke dalam purwarupa aplikasi web. Keterbatasan utama dalam penelitian ini terletak pada volume dataset latih yang sangat minim, yakni hanya berjumlah 45 citra kerusakan jalan, serta tingginya variasi gangguan visual berupa *noise* dan *blur* yang diterapkan saat augmentasi. Keterbatasan ketersediaan data ini berakibat fatal, di mana model YOLOv8 pada akhirnya mengalami *overfitting* karena terlalu sensitif menghafal pola data yang sempit sehingga kehilangan kemampuan generalisasi. Lebih lanjut, keterbatasan data ini juga menyebabkan algoritma SSD *MobileNet* gagal mencapai titik konvergensi, dengan metrik mAP yang terus stagnan di angka 30 persen selama 100 *epoch* berturut-turut. Kegagalan model SSD *MobileNet* ini menegaskan bahwa algoritma tersebut sangat membutuhkan keragaman fitur citra yang tinggi agar mekanisme *multi-feature maps* dapat bekerja secara optimal. Oleh karena itu, sebagai kemungkinan perbaikan dan saran untuk penelitian selanjutnya, sangat direkomendasikan untuk melakukan penambahan kuantitas sampel citra aspal rusak secara masif dan bervariasi. Perbaikan ketersediaan volume data ini sebaiknya difokuskan pada pengoptimalan algoritma SSD-*MobileNet*. Apabila dilatih dengan jumlah data yang memadai, SSD-*MobileNet* diyakini mampu mencapai tingkat akurasi yang tinggi sekaligus memberikan efisiensi operasional yang jauh lebih baik dari segi konsumsi RAM dan ruang penyimpanan, sehingga akan sangat ideal dan optimal untuk

diintegrasikan pada sistem pendeteksi di perangkat dengan sumber daya

UCAPAN TERIMA KASIH

Penulis mengucapkan banyak terimakasih kepada pihak-pihak yang telah membantu dalam pembuatan penelitian jurnal ini.

REFERENSI

- [1] E. Yulindra *et al.*, "Analisis Kerusakan Jalan Menggunakan Metode Bina Marga dan Metode Pavement Condition Index (PCI) (Studi Kasus : Jalan Sudirman dan Jalan Soekarno-Hatta Kota Dumai)," 2022.
- [2] M. Akbar, H. Betaubun, C. Utary, D. L. Pamuttu, and D. A. Pasalli, "1161-File Utama Naskah-5853-1-10-20231025," *JOURNAL OF RESEARCH AND INOVATION IN CIVIL ENGINEERING AS APPLIED SCIENCE (RIGID)*, vol. 2, pp. 7–13, Jan. 2023.
- [3] U. Shodiq *et al.*, "JOISIE licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0) KLASIFIKASI JALAN RUSAK MENGGUNAKAN TRANSFER LEARNING ARSITEKTUR VGG16," *Journal Of Information Systems And Informatics Engineering*, vol. 8, no. 1, pp. 75–85, 2024, doi: 10.35145/joisie.v8i1.4243.
- [4] K. A. Vinodhini and K. R. A. Sidhaarth, "Pothole detection in bituminous road using CNN with transfer learning," *Measurement: Sensors*, vol. 31, Feb. 2024, doi: 10.1016/j.measen.2023.100940.
- [5] N. Khairunisa, . C., and A. Jamaludin, "ANALISIS PERBANDINGAN ALGORITMA CNN DAN YOLO DALAM MENGIDENTIFIKASI KERUSAKAN JALAN," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, Aug. 2024, doi: 10.23960/jitet.v12i3.4434.
- [6] A. S. Dharma, C. N. S. Pardosi, and Z. P. Silaen, "Comparative Performance of Yolov8 and Ssd-mobilenet Algorithms for Road Damage Detection in Mobile Applications," *sinkron*, vol. 9, no. 3, pp. 1159–1169, Jul. 2025, doi: 10.33395/sinkron.v9i3.15008.
- [7] S. Hossain, H. Anzum, and S. Akhter, "Comparison of YOLO (V3,V5) and MobileNet-SSD (V1,V2) for Person Identification Using Ear-Biometrics," *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 1259–1271, Mar. 2024, doi: 10.12785/ijcds/150189.
- [8] S. N. Azzahra, D. Widhiantoro, K. Kunci, and D. Objek, "Analisa Komparatif Algoritma YOLO dan SSD untuk Sistem Deteksi Objek," *SEMINAR NASIONAL INOVASI VOKASI*, vol. 4, Jun. 2025.
- [9] A. I. Hanafi, A. Srirahayu, and A. Farida, "Sistem Rekomendasi Produk Konveksi Pada Deem Clothing Dengan Metode Knowledge Based," *SMATIKA JURNAL*, vol. 14, no. 02, pp. 270–283, Dec. 2024, doi: 10.32664/smatika.v14i02.1338.
- [10] "CRISP-DM: Tahapan, Studi Kasus, Kelebihan, dan Kekurangan." Accessed: Jun. 28, 2026. [Online]. Available: <https://www.dicoding.com/blog/crisp-dm-tahapan-studi-kasus-kelebihan-dan-kekurangan/>
- [11] D. B. Saputra, V. Atina, F. E. Nastiti, and F. I. Komputer, "PENERAPAN MODEL CRISP-DM PADA PREDIKSI NASABAH KREDIT MENGGUNAKAN ALGORITMA RANDOM FOREST," 2024. [Online]. Available: <http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/indexDwiBagusSaputra><http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index>
- [12] A. S. Alenezi and M. A. Mangoud, "Evaluating YOLOv8 for Road Damage Detection," Mar. 2025.
- [13] N. S. Putra, B. F. Hutabarat, and U. Khaira, "Implementasi Algoritma Convolutional Neural Network Untuk Identifikasi Jenis Kelamin Dan Ras," *Decode: Jurnal Pendidikan Teknologi Informasi*, vol. 3, no. 1, pp. 82–93, Feb. 2023, doi: 10.51454/decode.v3i1.123.
- [14] 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 (Switzerland)*, vol. 16, no. 24, Dec. 2024, doi: 10.3390/su162410964.
- [15] M. R. Albasith and M. Fachrie, "Real-Time Identification of Potholes and Road Damage Using Yolov11," *bit-Tech*, vol. 8, no. 2, pp. 2792–2802, Dec. 2025, doi: 10.32877/bt.v8i2.3371.
- [16] K. R. Ahmed, "Smart pothole detection using deep learning based on dilated convolution," *Sensors*, vol. 21, no. 24, Dec. 2021, doi: 10.3390/s21248406.
- [17] ul Hamidani, R. Yanto, S. Informasi, and S. Bina Nusantara Jaya Lubuklinggau, "Sistem Cerdas Makanan Sehat Metode CBR dan SMART untuk Edukasi Pemenuhan Gizi Masyarakat," *Jurnal Pustaka AI*, vol. 5, no. 2, pp. 355–363, 2025, doi: 10.55382/jurnalpustakaai.v5i2.1209.
- [18] "Pengertian dan Cara Kerja Algoritma Convolutional Neural Network (CNN) - Trivusi." Accessed: Jul. 11, 2026. [Online]. Available: <https://www.trivusi.web.id/2022/04/algoritma-cnn.html>
- [19] R. Liusman Gaho, I. T. Ali, and E. Prakasa, "Klasifikasi Kualitas Permukaan Jalan Raya Menggunakan Metode CNN Berbasis Arsitektur Xception," *JURNAL INOVTEK POLBENG - SERI INFORMATIKA*, vol. 9, no. 1, p. 2024, 2024.
- [20] H. E. Nawawi, R. Rahmadewi, and L. Nurpulaela, "Pendeteksian Wajah Hasil Dari Manipulasi Ai Pada Citra Digital," *Jurnal Ilmiah Wahana Pendidikan*, vol. 10, no. 2, pp. 402–409, 2024, doi: 10.5281/zenodo.10487513.
- [21] F. A. Rafi, A. Faggidae, and Y. T. Polly, "ASPHALT ROAD DAMAGE DETECTION SYSTEM USING CANNY EDGE DETECTION," *Jurnal Komputer dan Informatika*, vol. 11, no. 1, pp. 85–90, Mar. 2023, doi: 10.35508/jicon.v11i1.10100.
- [22] V. R. Yuniar, R. I. Qonata, H. A. Rizmadita, and A. S. Fitri, "PERANCANGAN SISTEM PELAPORAN KERUSAKAN JALAN DENGAN METODE OBJECT ORIENTED ANALYSIS DESIGN (OOAD) BERBASIS WEB," 2024.
- [23] J. Aplikasi Sains *et al.*, "Performasi Deteksi Jumlah Manusia Menggunakan YOLOv8," 2023. [Online]. Available: <https://universe.roboflow.com/csgo-head-detection/head-datasets>