



Deteksi dan Klasifikasi Kerusakan Pada Jalan Aspal Berbasis Citra Menggunakan Metode YOLO

Tugas Akhir

**Oleh:
Farhan Muhammad Yusuf (4212201074)**

**Program Studi Teknik Mekatronika
Jurusan Teknik Elektro
Politeknik Negeri Batam
2026**

Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul : “Deteksi dan Klasifikasi Kerusakan Pada Jalan Aspal Berbasis Citra Menggunakan Metode Yolo” adalah hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.



Farhan Muhammad Yusuf
NIM: 4212201074

Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Terapan Teknik (S.Tr.T)
di
Politeknik Negeri Batam

oleh:
Farhan Muhammad Yusuf (4212201074)

Tanggal Sidang: 20 Januari, 2026

Disetujui oleh :



1. Muhammad Naufal Airlangga
Diputra, S.Pd., M.P.H
NIK: 122281

1. Adlian Jefiza, S.Pd., M.T
NIK: 119220



2. Fadli Firdaus, M.Pd
NIK: 122271

Deteksi dan Klasifikasi Kerusakan Pada Jalan Aspal Berbasis Citra Menggunakan Metode YOLO

Abstrak

Kerusakan jalan aspal, khususnya retak buaya, merupakan permasalahan yang dapat menurunkan kenyamanan dan keselamatan pengguna jalan sehingga diperlukan sistem deteksi dan klasifikasi yang cepat dan akurat. Penelitian ini bertujuan untuk mengembangkan sistem deteksi dan klasifikasi kerusakan jalan aspal berbasis deep learning dengan mengintegrasikan metode You Only Look Once (YOLO) sebagai pendeteksi retakan dan Convolutional Neural Network (CNN) sebagai pengklasifikasi tingkat keparahan kerusakan. Model YOLOv12n digunakan untuk mendeteksi area retakan pada permukaan jalan, sedangkan model CNN dengan arsitektur ResNet50 digunakan untuk mengklasifikasikan kerusakan ke dalam tiga kelas, yaitu ringan, sedang, dan berat. Evaluasi kinerja model YOLO dilakukan menggunakan metrik mAP50, sementara evaluasi model CNN menggunakan confusion matrix dengan metrik akurasi, presisi, recall, dan F1-score. Hasil pengujian menunjukkan bahwa model YOLOv12n mampu mencapai nilai mAP50 sebesar 90,08% dan mampu mendeteksi retakan secara stabil, termasuk pada pengujian realtime. Model CNN ResNet50 menghasilkan akurasi sebesar 95,24% dengan performa yang seimbang pada seluruh kelas kerusakan. Kedua model kemudian diintegrasikan ke dalam aplikasi berbasis antarmuka yang mampu menampilkan hasil deteksi dan klasifikasi secara realtime melalui webcam dengan tampilan yang responsif dan stabil. Berdasarkan hasil penelitian, sistem yang dikembangkan terbukti mampu melakukan deteksi dan klasifikasi kerusakan jalan aspal secara efektif dan berpotensi digunakan sebagai alat bantu inspeksi jalan secara otomatis dan realtime.

Kata Kunci: Deteksi, Kerusakan Jalan, YOLO, CNN, ResNet-50, Klasifikasi, Jalan Aspal.

Image-Based Asphalt Road Damage Detection and Classification Using the YOLO Method

Abstract

Damage to asphalt roads can cause various problems, such as accidents, discomfort for road users, and increased repair costs if not addressed promptly. Current manual monitoring systems are still inefficient, thus requiring an automatic method capable of accurately detecting and classifying road damage. This study aims to develop an image-based system using the You Only Look Once (YOLO) method to detect asphalt road damage, and a Convolutional Neural Network (CNN) with ResNet-50 architecture to classify the severity levels of damage such as cracks, potholes, and deformations. The research process includes collecting and annotating a dataset of asphalt road images, image preprocessing, training the YOLO model as an object detector, and classifying the detection results using ResNet-50. Model performance is evaluated using metrics such as mean Average Precision (mAP), precision, recall, and F1-score. The developed system is expected to accurately recognize the types and severity levels of damage, thereby supporting automatic road condition monitoring and assisting in faster and more precise repair planning.

Keywords: Road Damage Detection, YOLO, CNN, ResNet-50, Classification, Asphalt Road.

Kata Pengantar

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini dengan baik. Laporan Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan jenjang Diploma IV pada Program Studi Teknik Mekatronika, Politeknik Negeri Batam.

Tugas Akhir ini membahas perancangan dan implementasi sistem deteksi serta klasifikasi kerusakan pada jalan aspal berbasis citra digital dengan memanfaatkan metode YOLO. Penelitian ini bertujuan untuk menghasilkan sistem yang mampu mengidentifikasi dan mengelompokkan tingkat kerusakan jalan secara otomatis, sehingga dapat membantu proses inspeksi dan pemantauan kondisi jalan menjadi lebih efektif, cepat, dan akurat. Sistem yang dikembangkan diharapkan dapat menjadi solusi pendukung dalam pengambilan keputusan terkait perencanaan perbaikan dan pemeliharaan infrastruktur jalan..

Dalam penyusunan Laporan Tugas Akhir ini, penulis menyadari bahwa keberhasilan penyelesaian laporan ini tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas segala nikmat, kesehatan, dan kesempatan yang di berikan.
2. Kedua orang tua dan keluarga, atas doa dan dukungannya.
3. Bapak Diono, S.T., M.Eng, selaku Kepala Program Studi Teknik Mekatronika.
4. Bapak Indra Hardian Mulyadi, S.T., M.Eng, selaku Kepala Jurusan Teknik Elektro.
5. Bapak Adlian Jefiza, S.Pd, M.T, selaku dosen pembimbing tugas akhir yang telah memberikan arahan dan bimbingan selama proses pengarjaan laporan akhir ini.
6. Seluruh dosen Teknik Elektro, atas ilmu dan pengalaman yang telah diberikan selama proses perkuliahan.
7. Teman-teman Teknik Mekatronika, yang telah memberikan dukungan, bantuan, dan kebersamaan selama proses penyusunan buku laporan akhir ini.

Batam, 20 Januari 2026



Farhan Muhammad Yusuf

Daftar Isi

Pernyataan Keaslian Tugas Akhir	i
Lembar Pengesahan	ii
Abstrak	iii
<i>Abstract</i>	iv
Kata Pengantar	v
Daftar Isi	vi
Daftar Gambar	ix
Daftar Tabel	x
Bab 1. Pendahuluan	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	2
1.4. Manfaat	3
1.5. Batasan	3
Bab 2. Tinjauan Pustaka	4
2.1. Jalan	4
2.1.1. Kerusakan Jalan	4
2.1.2. Retak Buaya (<i>Alligator Cracking</i>)	5
2.2. Deep learning	7
2.3. YOLO	8
2.5. CNN (Convolutional Neural Network)	9
2.4.1. Algoritma Optimizer Adam	10
2.5. Arsitektur Res Net50	11
2.6. <i>Preprocessing</i> Citra	12
2.6.1. Anotasi Citra	12
2.6.2. Pemotongan Area Kerusakan (<i>Crop</i>)	12
2.6.3. <i>Resize Image</i>	12

2.7. Augmentasi	13
2.8. Penelitian Terkait	14
2.9. Analisis Dan Gap Penelitian	17
Bab 3. Metodologi Penelitian	19
3. Perancangan.....	19
3.1. Perancangan Penelitian	19
3.1.1. Studi Literatur	20
3.1.2 Pengumpulan Dataset	20
3.1.3. Anotasi Data	21
3.1.4. <i>Preprocessing</i>	21
3.1.5. Pelatihan Model Yolo	22
3.1.6. Klasifikasi Tingkat Kerusakan Dengan CNN.....	22
3.1.7. Evaluasi Model	23
3.1.8. Kesimpulan	24
3.2. Alat dan Bahan.	25
3.3. Pengujian ukuran citra	25
3.4. Pengujian CNN Dengan Fungsi Aktivasi Yang Berbeda.	26
3.5. Pengujian CNN Klasifikasi Dengan Jumlah Epoch yang Berbeda.	27
3.6. Pengujian Dengan Learning Rate Yang Berbeda.....	27
3.7. Pengujian Dengan Data Tambahan	28
3.8. Analisis Pengujian.....	28
Bab 4. Hasil dan Pembahasan	30
4.1 Pengambilan Data	30
4.2 Anotasi Data	31
4.3 <i>Preprocessing</i> Citra.....	31
4.3.1. <i>Preprocessing</i> Citra Untuk YOLO.....	31
4.3.2. <i>Resize image</i> YOLO	32
4.3.3 <i>Preprocessing</i> Citra Untuk CNN	33
4.3.4 <i>Cropping</i> Citra.....	33

4.3.5 <i>Resize image</i> CNN	34
4.4. Pengujian Model Deteksi YOLO	35
4.4.1. Hasil Pengujian Variasi Ukuran Citra Pada Model YOLO	35
4.5. Pengujian Model Klasifikasi CNN	36
4.5.1. Hasil Pengujian Variasi Ukuran Citra Pada Model CNN	36
4.5.2. Hasil Pengujian Variasi Fungsi Aktivasi Pada Model CNN	38
4.5.3. Hasil Pengujian Variasi Jumlah Epoch	39
4.5.3. Hasil Pengujian Variasi <i>Learning Rate</i>	40
4.5.4. Evaluasi Model Akhir CNN	42
4.5.5. Hasil Pengujian Dengan Data Tambahan	44
4.6. Implementasi Model Pada GUI	47
4.6.1. Hasil Pengujian Fungsional GUI	48
4.6.1 Hasil Pengujian Secara <i>Realtime</i>	50
Bab 5. Kesimpulan dan Saran	57
5.1. Kesimpulan	57
5.2. Saran	57
Daftar Pustaka	59
Biodata	62
Lampiran	63

Daftar Gambar

Gambar 1. Retak Buaya (Ringan).....	5
Gambar 2. Retak Buaya (Sedang).....	5
Gambar 3. Retak Buaya (Berat).....	6
Gambar 4. Struktur <i>Deep Learning</i> [10].....	7
Gambar 5. Cara kerja YOLO [13].....	8
Gambar 6. Struktur CNN.....	9
Gambar 7. Arsitektur ResNet50	11
Gambar 8. Augmentasi Data	14
Gambar 9. Flowchart Perancangan Penelitian	19
Gambar 10. Contoh Pengambilan Gambar.....	30
Gambar 11. Anotasi Menggunakan Roboflow.....	31
Gambar 12. Penerapan Ukuran Citra Model YOLO	32
Gambar 13. Citra Hasil <i>Cropping</i>	33
Gambar 14. Penerapan Ukuran Citra Model CNN	34
Gambar 15. Grafik <i>Training</i> dan <i>Validation Loss</i> serta <i>Accuracy</i> pada Epoch 30 ..	40
Gambar 16. Grafik <i>Training</i> dan <i>Validation Loss</i> Pada <i>Learning Rate</i> 0,0001	41
Gambar 17. Confussion Matrix.....	43
Gambar 18. Tampilan GUI	48
Gambar 19. Hasil Deteksi dan Klasifikasi Realtime Kelas Berat	50
Gambar 20. Hasil Deteksi dan Klasifikasi Realtime Kelas Sedang	51
Gambar 21. Hasil Deteksi dan Klasifikasi Realtime Kelas Ringan.....	51
Gambar 22. Hasil Pengujian Pada Kondisi Pencahayaan Sore Hari	53
Gambar 23. Hasil Pengujian Pada Jalan Lubang	54
Gambar 24. Hasil Deteksi Pada Retakan Berukuran Kecil	55
Gambar 25. Hasil Deteksi Pada Retakan Berukuran Kecil	55

Daftar Tabel

Tabel 1. Penelitian Mengenai Kerusakan Jalan	14
Tabel 2. Estimasi biaya	25
Tabel 3. Pengujian Ukuran Citra YOLO	25
Tabel 4. Pengujian Ukuran Citra CNN	26
Tabel 5. Pengujian Fungsi Aktivasi CNN	26
Tabel 6. Pengujian dengan Jumlah Epoch yang Berbeda	27
Tabel 7. Pengujian Dengan <i>Learning Rate</i>	27
Tabel 8. Pengujian Dengan Data Tambahan	28
Tabel 9. Analisis.	28
Tabel 10. Hasil Pengujian Ukuran Citra	35
Tabel 11. Hasil Pengujian Ukuran Citra CNN	37
Tabel 12. Hasil Pengujian Fungsi Aktivasi	38
Tabel 13. Hasil Pengujian Variasi Jumlah Epoch	39
Tabel 14. Hasil Pengujian <i>Learning Rate</i>	41
Tabel 15. Ringkasan Kinerja Model CNN	42
Tabel 16. Hasil Evaluasi Klasifikasi Model CNN pada Data Uji	43
Tabel 17 Hasil Pengujian Model CNN Menggunakan Data Tambahan	45
Tabel 18. Hasil Pengujian Fungsional	48
Tabel 19. Hasil Pengujian Gui Secara Realtime	52

Bab 1. Pendahuluan

1.1. Latar Belakang

Kerusakan jalan aspal merupakan permasalahan umum yang sering terjadi di berbagai wilayah Indonesia, terutama pada jalan-jalan dengan volume lalu lintas tinggi dan berada di daerah beriklim tropis lembap. Jenis kerusakan seperti retak, lubang, dan deformasi permukaan tidak hanya menurunkan kenyamanan berkendara, tetapi juga meningkatkan risiko kecelakaan serta menimbulkan biaya perawatan yang cukup besar. Berdasarkan data dari Kementerian Pekerjaan Umum dan Perumahan Rakyat (PUPR) tahun 2023, dari total sekitar 47.604 km jalan nasional non-tol, sebanyak 5,8% atau sekitar 2.770 km berada dalam kondisi tidak mantap, yaitu rusak ringan hingga rusak berat. Selain itu, tercatat terdapat sekitar 3.500 km ruas jalan nasional yang mengalami kerusakan dan belum tertangani secara optimal. Kondisi ini tentu berdampak langsung terhadap keselamatan pengguna jalan serta efisiensi distribusi logistik nasional.[1]

Metode *You Only Look Once* (YOLO) adalah algoritma deteksi objek yang menggunakan teknik *deep learning* dan mampu melakukan pendeteksian secara *real-time* dengan tingkat akurasi yang tinggi. YOLO mengintegrasikan proses deteksi dan klasifikasi ke dalam satu jaringan saraf tunggal, sehingga mempercepat waktu pemrosesan tanpa mengurangi ketepatan hasil.[2] Dalam aplikasi untuk mendeteksi kerusakan jalan, beberapa varian YOLO seperti YOLOv8 dan YOLO 12 telah dikembangkan dengan tambahan modul khusus yang meningkatkan kemampuan dalam mengenali kerusakan yang kecil, rumit, serta berukuran berbeda pada permukaan jalan. Pendekatan ini memberikan hasil identifikasi kerusakan yang lebih tepat dan efisien dibandingkan dengan metode konvensional.

Beberapa penelitian di Indonesia telah memanfaatkan metode YOLO untuk mendeteksi kerusakan jalan dengan hasil yang cukup memuaskan. Namun, masih ada sejumlah keterbatasan yang menjadi celah penelitian. Sebagai contoh, penelitian yang dilakukan oleh Dimensi PPI Petra mengembangkan sistem inspeksi otomatis berbasis YOLO yang dapat mendeteksi kerusakan jalan secara *real-time* dengan akurasi yang tinggi. Namun, sistem tersebut masih membutuhkan perangkat keras tambahan dan belum sepenuhnya mandiri untuk aplikasi di lapangan secara luas[3]. Sementara itu, studi dari Universitas Diponegoro menemukan bahwa meskipun YOLO cukup efektif dalam mendeteksi kerusakan jalan, masih ada ruang untuk meningkatkan akurasi dan efisiensi komputasi, terutama jika digunakan dalam skala besar dan secara *real-time*[4].

Namun demikian, sebagian besar penelitian tersebut masih fokus pada proses deteksi keberadaan kerusakan saja, belum sampai pada klasifikasi tingkat keparahan kerusakan, yang sangat penting untuk penentuan prioritas perbaikan

jalan. Oleh karena itu, penelitian ini tidak hanya menggunakan YOLOv12 untuk mendeteksi lokasi kerusakan, tetapi juga mengintegrasikan metode Convolutional Neural Network (CNN) untuk mengklasifikasikan tingkat keparahan kerusakan, seperti ringan, sedang, dan berat, khususnya pada jenis kerusakan retak buaya. CNN sebagai salah satu arsitektur *deep learning* yang efektif dalam pengolahan citra, digunakan untuk mengenali pola dan tekstur retakan pada hasil deteksi YOLO secara lebih mendalam.

Melalui kombinasi pendekatan ini, penelitian ini bertujuan untuk menghasilkan sistem yang tidak hanya mampu mendeteksi lokasi kerusakan jalan secara akurat, tetapi juga memberikan informasi mengenai tingkat keparahannya. Dengan demikian, sistem ini diharapkan dapat menjadi solusi yang efisien dan handal dalam mendukung pengelolaan pemeliharaan jalan yang lebih sistematis dan berkelanjutan.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan berikut merupakan rumusan masalah dalam penelitian ini.:

1. Bagaimana metode *You Only Look Once* (YOLO) dapat diimplementasikan dan dianalisis performanya dalam mendeteksi kerusakan pada permukaan jalan aspal?
2. Bagaimana metode Convolutional Neural Network (CNN) dapat diimplementasikan dan dianalisis akurasi dalam mengklasifikasikan tingkat keparahan kerusakan pada jalan aspal?

1.3. Tujuan

Penelitian ini memiliki beberapa tujuan utama yang hendak di capai, dijelaskan sebagai berikut:

1. Mengimplementasikan dan menganalisis performa metode *You Only Look Once* (YOLO) dalam mendeteksi kerusakan pada permukaan jalan aspal.
2. Mengimplementasikan dan menganalisis akurasi metode Convolutional Neural Network (CNN) dalam mengklasifikasikan tingkat keparahan kerusakan pada jalan aspal.

1.4. Manfaat

Pada bagian ini di paparkan kontribusi yang diharapkan dari penelitian manfaat bagi pihak tertentu maupun peneliti sendiri.

1. Penelitian ini dapat berfungsi sebagai sarana pendukung dalam perawatan jalan untuk pihak pengelola kondisi jalan seperti Binamarga atau Dinas Pekerjaan Umum (DPU).

1.5. Batasan

Agar Tugas akhir ini lebih terarah dan sesuai, maka harus dibatasi dengan hal-hal berikut:

1. Penelitian ini berfokus pada kerusakan jalan aspal jenis retak buaya (*alligator cracking*).
2. Deteksi lokasi kerusakan dilakukan menggunakan metode YOLO (*You Only Look Once*).
3. Klasifikasi tingkat keparahan (ringan, sedang, berat) dilakukan menggunakan Convolutional Neural Network (CNN).
4. Tidak mencakup jenis kerusakan lain seperti lubang, retakan memanjang, atau kerusakan bentuk lainnya.
5. Citra yang digunakan merupakan foto permukaan jalan dengan fokus pada area retak buaya.
6. Gambar diambil dari sudut yang memperjelas kerusakan, dalam kondisi pencahayaan baik untuk menjamin kualitas data visual dalam proses pelatihan dan inferensi.

Bab 2. Tinjauan Pustaka

Pada bagian tinjauan Pustaka ini akan diuraikan pengenalan teori menjadi dasar bagi penelitian. Hal ini mencakup definisi serta klasifikasi kerusakan jalan aspal, prinsip kerja metode YOLO yang digunakan untuk deteksi objek serta algoritma Covolutional Neural Network (CNN) yang digunakan untuk proses klasifikasi tingkat kerusakan jalan.

2.1. Jalan

Berdasarkan Peraturan Pemerintah Nomor 34 Tahun 2006 Jalan adalah fasilitas transportasi darat yang mencakup seluruh bagian jalan beserta bangunan pelengkap dan perlengkapannya yang digunakan untuk lalu lintas. Jalan tersebut dapat terletak di permukaan tanah, di atas atau di bawah tanah maupun air, termasuk juga yang berada di atas permukaan air. Namun, pengertian ini tidak mencakup jalan kereta api, jalan lori, dan jalan kabel [5]. Fungsi jalan adalah sebagai penghubung antar wilayah dan membantu kelancaran kegiatan sosial dan ekonomi masyarakat.

2.1.1. Kerusakan Jalan

Dalam dunia transportasi, khususnya pada infrastruktur jalan raya, kondisi jalan bisa mengalami berbagai jenis kerusakan seiring bertambahnya waktu. Umumnya, kerusakan ini disebabkan oleh beban kendaraan yang terlalu berat, cuaca ekstrem seperti hujan deras atau panas terik, serta umur jalan yang sudah lama digunakan tanpa pemeliharaan rutin [6]. Jenis-jenis kerusakan jalan biasanya diklasifikasikan berdasarkan bentuk dan faktor penyebabnya. Di lapangan, kita sering menjumpai beberapa bentuk kerusakan umum seperti retakan, dan lubang.

Penelitian ini secara khusus memfokuskan objek deteksi dan klasifikasi pada jenis kerusakan jalan berupa retak buaya (*alligator cracking*) karena beberapa alasan teknis dan praktis. Retak buaya merupakan indikator awal dari kerusakan struktural serius pada perkerasan jalan aspal, yang biasanya disebabkan oleh beban lalu lintas berulang atau kualitas fondasi jalan yang menurun. Retakan ini berbentuk seperti jaringan atau pola sisik kulit buaya, dan umumnya berkembang dari retak halus hingga menyebabkan pecahan permukaan jalan jika tidak segera ditangani[7]

Beberapa studi menyebutkan bahwa retak buaya sering diabaikan dalam sistem deteksi otomatis, yang cenderung hanya fokus pada kerusakan yang lebih besar dan mencolok seperti lubang jalan (*pothole*). Padahal, deteksi dini terhadap retak buaya sangat krusial untuk menghindari biaya perbaikan yang lebih besar dan meningkatkan umur layanan perkerasan. Sebuah penelitian oleh Sari melaporkan bahwa metode segmentasi seperti Active Contour Model masih

mengalami kesalahan klasifikasi terutama pada retak buaya, karena pola dan strukturnya yang kompleks.[8]

2.1.2. Retak Buaya (*Alligator Cracking*)

Retak ini memiliki pola seperti jaring yang mirip dengan kulit buaya, dengan lebar retakan minimal 3 mm. Umumnya, retak ini muncul akibat beban lalu lintas yang berlebihan dan berlangsung secara terus-menerus. Selain itu, kemungkinan penyebab lainnya berasal dari proses konstruksi, seperti penggunaan material perkerasan yang kurang berkualitas sehingga membuat struktur jalan mudah rapuh dan aspal mengalami pelapukan.

Tingkat kerusakan retak buaya dibagi menjadi tigakelas utama sebagai berikut:

- Ringan
Retak tipis atau halus yang memanjang sejajar satu sama lain, bisa saling terhubung atau tidak, dan tidak disertai dengan penggumpalan.



Gambar 1. Retak Buaya (Ringan)

- Sedang
Retak kulit buaya ringan berkembang membentuk pola atau jaringan retakan, yang kemudian disertai dengan gumpalan ringan.



Gambar 2. Retak Buaya (Sedang)

- Berat

Pola dan jaringan retakan telah berkembang lebih lanjut, sehingga pecahan-pecahan permukaan terlihat jelas, disertai gompal di bagian tepinya. Beberapa pecahan menunjukkan pergerakan (*rocking*) akibat beban lalu lintas.[9]

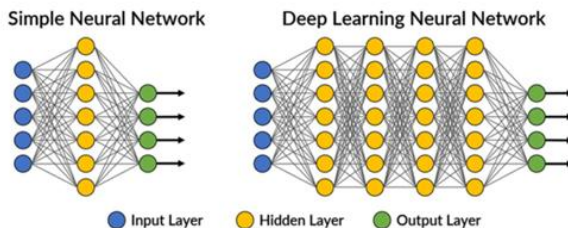


Gambar 3. Retak Buaya (Berat)

2.2. Deep learning

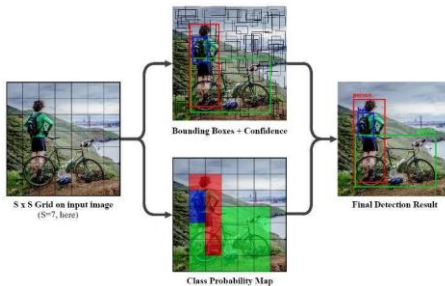
Deep Learning (DL) merupakan metode yang umum dipakai dalam beberapa tahun terakhir untuk mengaplikasikan *Machine Learning* (ML). Metode ini didasarkan pada jaringan saraf tiruan yang memiliki banyak lapisan, biasanya lebih dari tujuh[10]. Metode ini bekerja dengan meniru cara kerja otak manusia melalui jaringan neuron yang saling terhubung, sehingga memungkinkan sistem untuk memahami data seperti gambar, suara, dan teks tanpa perlu bantuan manusia dalam mengekstrak fitur secara manual. *Deep learning* saat ini banyak digunakan dalam berbagai bidang, termasuk deteksi objek, pengenalan suara, dan pengolahan bahasa alami, karena kemampuannya dalam mengelola data dalam jumlah besar dengan tingkat akurasi yang tinggi[11].

Secara garis besar, model *deep learning* terdiri dari tiga jenis lapisan utama, yaitu lapisan *deep*, lapisan tersembunyi (*hidden layer*), dan lapisan *output*. Lapisan *input* berperan sebagai penerima data mentah yang akan diproses lebih lanjut. Sementara itu, lapisan tersembunyi berfungsi untuk mengekstrak fitur serta mempelajari pola-pola yang terkandung dalam data tersebut. Jumlah lapisan tersembunyi ini bisa bervariasi, mulai dari yang sedikit (*shallow*) hingga sangat banyak (*deep*), tergantung pada tingkat kompleksitas model yang diperlukan. Lapisan *output* bertanggung jawab untuk menghasilkan prediksi akhir, seperti klasifikasi atau nilai regresi. Dengan bertambahnya jumlah lapisan tersembunyi, kemampuan model untuk mengenali pola-pola kompleks pada data besar dan tidak terstruktur juga meningkat, yang biasanya berdampak pada peningkatan akurasi. Oleh karena itu, *deep learning* sangat cocok digunakan untuk menyelesaikan masalah yang melibatkan data dalam jumlah besar dan bersifat tidak terstruktur, seperti pengolahan gambar, suara, maupun teks[12].



Gambar 4. Struktur *Deep Learning* [10]

2.3. YOLO



Gambar 5. Cara kerja YOLO [13]

YOLO (*You Only Look Once*) merupakan algoritma deteksi objek *real-time* yang bekerja dengan membagi citra menjadi grid dan memprediksi *bounding box* serta kelas objek secara langsung dalam satu tahap inferensi, sehingga menghasilkan proses yang cepat dan efisien. YOLO memformulasikan deteksi objek sebagai sebuah masalah regresi tunggal yang mengubah langsung piksel gambar menjadi koordinat *bounding box* serta probabilitas kelas objek. Dengan pendekatan ini, YOLO hanya memproses gambar satu kali saja untuk menentukan jenis objek yang ada dan lokasi tepatnya dalam gambar.

Algoritma YOLO sudah banyak dipakai dalam berbagai penelitian, terutama untuk mendeteksi kerusakan pada jalan. Salah satu penelitian tahun 2025 mengembangkan sistem deteksi lubang jalan secara *real-time* menggunakan YOLOv8 dan perangkat ESP32-CAM. Hasilnya cukup bagus dengan F1-score 0,73 dan waktu proses yang cepat, jadi terbukti YOLOv8 efektif untuk pengolahan gambar infrastruktur jalan[14]. Selain itu, ada juga penelitian yang memakai YOLOv11 dalam sistem berbasis web untuk mendeteksi lubang jalan secara *real-time* dengan akurasi 83,5% dan *recall* 92,4%, yang menunjukkan performa yang cukup baik dalam mengenali kerusakan jalan[15]. Penelitian lain juga menggunakan YOLO untuk mendeteksi dan mengklasifikasikan berbagai jenis kerusakan jalan dari gambar digital, dengan hasil akurasi tinggi dan proses deteksi yang lebih cepat dibanding metode konvensional.

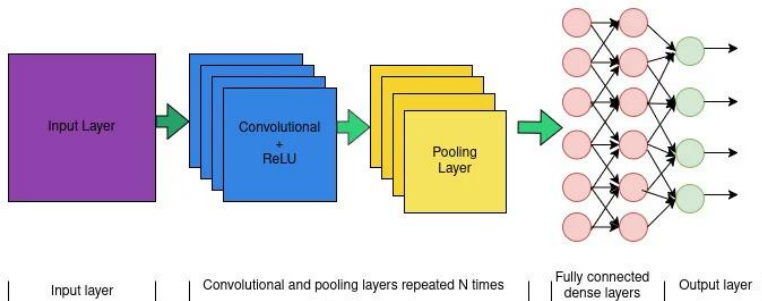
Sebagai perbandingan, algoritma Faster R-CNN merupakan metode two-stage detector yang dikenal dengan akurasi tinggi dalam mengenali objek dengan bentuk kecil dan kompleks, meskipun memiliki waktu proses yang lebih lama dibandingkan YOLO. Dalam konteks deteksi kerusakan jalan di Indonesia, penelitian oleh Fajar menggunakan Faster R-CNN untuk mendeteksi kerusakan

jalan di kawasan Pakansari, Bogor, dan berhasil mencapai akurasi sebesar 96,84%, meskipun kecepatan deteksinya hanya 0,16 gambar/detik[16].

Berdasarkan hasil tersebut, dapat disimpulkan bahwa YOLO sangat cocok untuk kebutuhan *real-time* dan efisiensi, sementara Faster R-CNN lebih unggul dalam akurasi namun kurang optimal jika diterapkan pada sistem yang membutuhkan kecepatan tinggi. Dengan mempertimbangkan kebutuhan sistem deteksi otomatis kerusakan jalan yang efisien dan cepat, algoritma YOLO menjadi pilihan yang tepat dalam penelitian ini.

2.4. CNN (Convolutional Neural Network)

Convolutional Neural Network (CNN) adalah salah satu metode dalam *deep learning* yang berkembang dari Multi layer perceptron (MLP), Khusus dirancang untuk olah data berbentuk grid. Misalnya citra dua dimensi seperti foto atau rekaman suara. CNN di kelompokkan sebagai *deep learning* karena memiliki banyak lapisan tersembunyi dan sering dimanfaatkan untuk memproses data gambar. Pada dasarnya pengklasifikasian gambar bisa dilakukan dengan (MLP), tetapi cara ini kurang efektif karena tidak mempertahankan informasi spasial dari data citra dan menganggap setiap piksel sebagai fitur yang berdiri sendiri, sehingga menghasilkan kinerja yang tidak maksimal.[17]



Gambar 6. Struktur CNN

Sumber = [Codecademy team](#)

Arsitektur CNN terdiri dari berbagai lapisan, masing-masing melakukan tugas tertentu untuk mengekstrak fitur dari gambar. CNN terdiri dari beberapa komponen utama yang meliputi *convolution layer*, *pooling layer*, dan *fully connected layer*.

1. *Convolution layer*

Convolution layer adalah lapisan pertama yang memproses data gambar. Fungsi utamanya adalah untuk menggunakan filter (atau kernel) untuk mengekstrak fitur dari gambar. Filter ini bergerak melintasi gambar dan melakukan operasi konvolusi untuk menghasilkan peta fitur, juga dikenal sebagai peta fitur. Setiap filter memiliki kemampuan untuk mendeteksi karakteristik tertentu, seperti tepi, sudut, atau tekstur.

2. *Pooling layer*

Setelah Convolution layer mengekstrak fiturnya, *pooling layer* digunakan untuk dimensi dan kompleksitas. *Max pooling*, yang mengambil nilai maksimum dari area kecil dalam peta fitur, atau *average pooling*, yang mengambil nilai rata-rata dari area kecil, adalah dua metode yang biasanya digunakan untuk melakukan pooling.

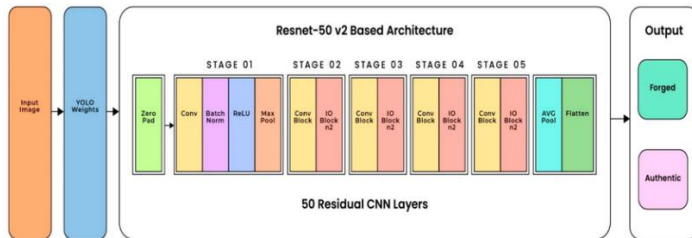
3. *Fully Connected Layer*

Setelah melewati beberapa lapisan sebelumnya, *output* dari fitur akan dimasukkan ke dalam *fully connected layer*. Pada lapisan ini neuron neuron terhubung satu sama lain, dan disinilah keputusan akhir di buat.[18]

2.4.1. Algoritma Optimizer Adam

Algoritma optimizer digunakan untuk meningkatkan akurasi, menemukan bobot optimal, dan mengurangi kesalahan. Selama proses pelatihan, parameter model, atau bobot, diubah untuk mencoba dan mengurangi kerugian agar dapat memprediksi seakurat mungkin. Pada penelitian ini menggunakan optimizer adam karena kemampuannya dalam mempercepat konvergensi dan memberikan hasil akurat serta stabil pada saat proses pelatihan model, khususnya untuk mengoptimalkan bobot agar model bisa memaksimalkan akurasi prediksi dan bisa mengurangi kesalahan.[19]

2.5. Arsitektur Res Net50



Gambar 7. Arsitektur ResNet50

Sumber = [ResearchGate](#)

ResNet adalah salah satu arsitektur dari *Convolutional Neural Network* (CNN) yang dirancang untuk melatih jaringan dengan kedalaman yang tinggi tanpa mengalami penurunan kinerja, sebuah masalah yang sering muncul pada jaringan yang sangat dalam.[20].

ResNet50 merupakan varian dari arsitektur ResNet yang memiliki total 50 lapisan, yang terdiri atas 48 lapisan konvolusi, satu lapisan *max pooling*, dan satu lapisan *average pooling*. Arsitektur ini menggunakan blok konvolusi dasar yang mencakup operasi konvolusi berukuran 3x3, normalisasi batch (*batch normalization*), serta fungsi aktivasi ReLU (*Rectified Linear Unit*). Selain itu, terdapat blok residual yang menjadi ciri khas dari ResNet. Blok residual ini memungkinkan jaringan untuk menangkap representasi fitur yang lebih kompleks secara lebih efisien, serta membantu mengatasi permasalahan degradasi performa saat jaringan menjadi semakin dalam[21].

Dibandingkan arsitektur lain seperti VGG-19 dan VGG-16, ResNet-50 menawarkan efisiensi parameter lebih baik dan pelatihan lebih cepat, sekaligus mempertahankan akurasi tinggi. Berdasarkan studi oleh Yanni dkk. dari Jurnal Ilmiah Informatika Komputer, arsitektur ResNet-50 mengungguli VGG-19 dalam tugas klasifikasi kerusakan jalan, dengan akurasi lebih baik dan waktu latih lebih singkat[22]. Dan juga dengan arsitektur lain seperti MobileNet-V2, ResNet-50 menawarkan ketahanan pelatihan yang lebih unggul dalam menangani pola tekstur yang rumit, meskipun MobileNet-V2 lebih ringan dan cepat di perangkat dengan keterbatasan komputasi. Namun, pada klasifikasi jenis kerusakan jalan atau klasifikasi multi-kelas seperti jenis buah kurma, penelitian oleh Agyl Restu Hermanto dkk. menunjukkan bahwa ResNet-50 memberikan hasil lebih stabil dan akurat dibanding MobileNet-V2 yang lebih efisien namun terkadang kurang sensitif terhadap variasi tekstur halus ketika jumlah data terbatas.[23]

Oleh karena itu, dalam penelitian ini dipilih arsitektur ResNet-50 karena kemampuannya dalam mengenali pola visual yang rumit secara efisien, serta telah

terbukti menghasilkan performa klasifikasi yang lebih unggul dibandingkan beberapa arsitektur CNN lainnya dalam tugas serupa

2.6. Preprocessing Citra

Preprocessing merupakan langkah awal yang krusial dalam pengolahan citra digital sebelum digunakan pada proses pelatihan model berbasis *deep learning*. Tujuan dari tahap ini adalah untuk mempersiapkan data agar memenuhi standar format, dimensi, dan skala nilai yang sesuai dengan kebutuhan model. Selain itu, *preprocessing* juga bertujuan untuk meningkatkan kualitas serta variasi data sehingga model dapat melakukan pembelajaran secara lebih optimal. Dalam penelitian ini, *preprocessing* dilakukan melalui beberapa tahapan berikut:

2.6.1. Anotasi Citra

Langkah pertama dalam proses *preprocessing* adalah pemberian anotasi pada citra untuk menandai area yang mengindikasikan adanya kerusakan pada permukaan jalan. Proses anotasi dilakukan dengan menggunakan perangkat lunak Roboflow, untuk membuat *bounding box* pada area kerusakan. Hasil anotasi disimpan dalam format YOLO (*You Only Look Once*), yaitu dalam bentuk berkas teks yang berisi informasi mengenai kelas dan koordinat *bounding box* dalam rasio terhadap dimensi citra.[24]

2.6.2. Pemotongan Area Kerusakan (Crop)

Setelah model YOLO mendeteksi objek kerusakan dalam citra, sistem akan menghasilkan koordinat dari area tersebut. Berdasarkan koordinat ini, citra kemudian dipotong pada bagian yang mengandung objek kerusakan. Citra hasil pemotongan ini akan menjadi masukan (*input*) proses klasifikasi menggunakan model Convolutional Neural Network (CNN). Hal ini dimaksudkan untuk memperkecil dimensi citra gambar dan memotong waktu komputasi.[25]

2.6.3. Resize Image

Proses ini dilakukan untuk menyamakan ukuran setiap citra dengan cara mengubah skala piksel, sehingga seluruh dataset memiliki dimensi yang konsisten. Hal ini menjadi penting karena model CNN dan YOLO hanya menerima *input* dengan ukuran tertentu. Oleh sebab itu, penyesuaian ukuran gambar perlu dilakukan agar sesuai dengan arsitektur model yang digunakan. [26]

2.7. Augmentasi

Pada penelitian ini, augmentasi data dilakukan sebagai bagian dari tahap *preprocessing* untuk meningkatkan keragaman citra retakan jalan aspal yang digunakan dalam pelatihan model. Tujuan utama dari proses ini adalah untuk memperluas representasi visual dari retakan, sehingga model CNN dapat belajar mengenali berbagai bentuk dan arah retakan tanpa mengalami *overfitting*. Dengan augmentasi, model dilatih agar lebih adaptif terhadap perbedaan tampilan yang mungkin muncul pada citra di kondisi nyata.[27]

Teknik augmentasi yang digunakan dalam penelitian ini meliputi rotasi acak, pembalikan arah horizontal dan vertikal, penyesuaian warna (*color jitter*), serta transformasi translasi. Rotasi acak hingga $\pm 20^\circ$ diterapkan untuk meniru orientasi retakan yang beragam di permukaan jalan. Pembalikan horizontal dengan probabilitas 0.5 dan pembalikan vertikal dengan probabilitas 0.3 digunakan untuk menghasilkan variasi arah dari pola retakan yang sama. Sementara itu, *color jitter* digunakan untuk menyesuaikan tingkat kecerahan (*brightness* 0.3), kontras (*contrast* 0.3), saturasi (*saturation* 0.3), dan *hue* (0.1), yang bertujuan merepresentasikan kondisi pencahayaan berbeda saat pengambilan citra di lapangan, seperti bayangan atau cahaya matahari langsung.

Selain itu, digunakan pula transformasi affine dengan translasi (0.1, 0.1) untuk memberikan efek pergeseran posisi secara acak pada citra. Transformasi ini membantu model agar lebih toleran terhadap variasi posisi dan sudut pengambilan gambar, yang sering kali berbeda antar citra di lapangan. Setelah proses augmentasi dilakukan, citra diubah ke bentuk tensor dan dinormalisasi menggunakan nilai rata-rata dan standar deviasi RGB yang umum digunakan pada model CNN pra-latih, yaitu mean [0.485, 0.456, 0.406] dan *standard deviation* [0.229, 0.224, 0.225].

Pada tahap validasi, citra tidak melalui proses augmentasi, melainkan hanya dilakukan *resize* dan normalisasi. Pendekatan ini bertujuan agar evaluasi performa model mencerminkan kemampuan sebenarnya tanpa dipengaruhi oleh variasi buatan dari proses augmentasi. Dengan kombinasi teknik augmentasi yang diterapkan pada data pelatihan, model CNN diharapkan mampu melakukan generalisasi yang lebih baik terhadap citra retakan dengan berbagai kondisi pencahayaan, orientasi, dan posisi. Hal ini diharapkan dapat meningkatkan akurasi dan ketahanan model dalam mendeteksi retakan jalan aspal pada kondisi nyata.



Gambar 8. Augmentasi Data

2.8. Penelitian Terkait

Tabel 1. Penelitian Mengenai Kerusakan Jalan

N o	Tahu n	Penulis	Jurnal	Metode	Kelemahan
1	2023	Sasmito dkk.	Deteksi Kerusakan Jalan Berdasarkan Citra Digital Menggunakan n CNN	CNN	Rentan <i>overfitting</i> , membutuhkan dataset besar dan beragam untuk hasil optimal.
2	2025	Tim Dimensi PPI Petra	Analisis Deteksi Kerusakan pada Jalan Aspal menggunakan n <i>Deep</i>	YOLO + CNN	Perlu data berlabel banyak; performa menurun pada citra buram dan

			<i>Learning</i> untuk Mendukung Efisiensi Biaya dan Waktu		kondisi pencahayaan buruk
3	2022	Muhammad Mahrus Ali.	Deteksi Jalan Berlubang Menggunakan Metode Grey Level Co- Occurrence Matrix dan Neural Network	Ekstraksi fitur GLCM + Klasifikasi Neural Network (NN)	Precision rendah (0,66); dataset terbatas (29 citra); error rate 0,20; hasil deteksi kurang optimal terutama pada beberapa citra
4	2022	Kumar & Kumar	<i>Pothole Detection System Using YOLO v4 Algorithm</i>	<i>Preprocessing</i> + YOLOv4	Tidak mengklasifikasi jenis kerusakan
5	2024	Muhammad Surahmanto dkk	Deteksi Jalan Berlubang Menggunakan Algoritma YOLOv5	YOLOv5	Akurasi tinggi, namun performa menurun pada kondisi pencahayaan buruk dan

					video dengan gerakan cepat
6	2024	Erwin Syahrudin dkk.	<i>Enhanced YOLOv8 with OpenCV for Blind-Friendly Object Detection and Distance Estimation</i>	YOLOv8 + OpenCV	Dataset terbatas, hanya 1 jenis kerusakan
7	2021	Yuslena Sari dkk.	Penerapan Active Contour Model Pada Pengolahan Citra Untuk Deteksi Kerusakan Jalan.	Segmentasi Active Contour Model + Klasifikasi SVM	Akurasi 87,93%, kesalahan klasifikasi masih terjadi terutama pada jenis retak kulit buaya (<i>alligator cracking</i>)
8	2025	Royhan E. (Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer)	Sistem Deteksi Lubang Jalan <i>Real-time</i> dengan YOLOv8 dan ESP32-CAM	YOLOv8 + <i>Embedded System</i>	Performa <i>real-time</i> baik, tapi perangkat embedded memiliki keterbatasan sumber daya dan pencahayaan

9	2025	Mulyana D dkk	Deteksi Kerusakan Jalan Berdasarkan Citra Digital Menggunakan Convolutional Neural Network (CNN)	CNN	Fokus penelitian terbatas pada klasifikasi kerusakan jalan berdasarkan citra digital tanpa implementasi langsung pada video atau sistem monitoring <i>real-time</i> .
10	2023	Nirmal Kumar dkk	<i>Improved Pothole Detection Using YOLOv7 and ESRGAN</i>	YOLOv7 + super-resolution (ESRGAN) untuk deteksi <i>pothole</i> dari gambar <i>dashcam</i>	Fokus ke <i>pothole</i> saja; belum untuk retak tipis atau klasifikasi keparahan

2.9. Analisis Dan Gap Penelitian

Berdasarkan studi literatur yang telah dilakukan, sebagian besar penelitian terkait deteksi kerusakan jalan menggunakan pendekatan CNN, YOLO, ataupun metode klasifikasi tradisional seperti GLCM dan Neural Network. Namun, dari berbagai studi tersebut, terdapat beberapa kelemahan umum yang membuka celah (*gap*) untuk penelitian lebih lanjut, yaitu:

1. Fokus terbatas pada jenis kerusakan tertentu: Banyak penelitian hanya menargetkan lubang jalan (*pothole*), sementara jenis retakan permukaan seperti retak buaya (*alligator cracking*) yang lebih kompleks jarang dijadikan fokus utama.

2. Tidak adanya klasifikasi tingkat keparahan: Sebagian besar studi hanya melakukan deteksi keberadaan kerusakan, tanpa membedakan tingkat keparahannya (ringan, sedang, berat), padahal informasi ini sangat penting untuk prioritas perbaikan jalan.
3. Keterbatasan dataset: Beberapa penelitian masih menggunakan jumlah data terbatas atau hanya mengandalkan satu jenis kerusakan, sehingga model cenderung kurang general dan tidak *robust* terhadap kondisi nyata di lapangan.
4. Kinerja terganggu oleh pencahayaan dan gerakan kamera: Studi seperti oleh Surahmanto. dan Royhan E. menunjukkan bahwa performa YOLO dapat menurun dalam kondisi pencahayaan buruk atau video bergerak cepat, terutama jika tidak dilakukan augmentasi dan *preprocessing* yang tepat.
5. Tidak ada klasifikasi spesifik untuk retak buaya: Penelitian yang sudah ada belum secara spesifik melakukan deteksi dan klasifikasi retak buaya, padahal retakan ini merupakan indikator awal kerusakan struktural serius.

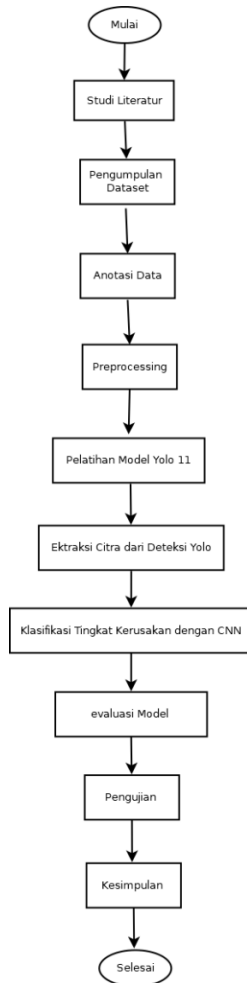
Tugas akhir ini berusaha mengatasi beberapa *gap* tersebut dengan:

1. Deteksi objek menggunakan YOLOv12 Untuk mengidentifikasi lokasi retakan buaya secara cepat dan akurat dari citra jalan.
2. Klasifikasi tingkat keparahan menggunakan ResNet-50 Setelah objek retakan terdeteksi, gambar potongan (*patch*) dari retakan akan diklasifikasikan ke dalam tiga kelas keparahan: ringan, sedang, dan berat menggunakan arsitektur ResNet-50 yang andal dalam ekstraksi fitur tekstur dan pola.
3. Melakukan analisis terhadap hasil pengujian.
4. Melakukan evaluasi menyeluruh menggunakan *accuracy*, *precision*, *recall*, dan *f1-score*.

Bab 3. Metodologi Penelitian

3. Perancangan

3.1. Perancangan Penelitian



Gambar 9. Flowchart Perancangan Penelitian

3.1.1. Studi Literatur

Studi literatur dilakukan untuk mendapatkan pemahaman yang lebih dalam mengenai teknologi yang digunakan dalam penelitian ini, khususnya terkait penerapan computer vision dalam mendeteksi dan mengklasifikasi retakan buaya pada permukaan jalan aspal. Tahap ini dilakukan dengan mencari dan mempelajari berbagai sumber ilmiah, seperti jurnal, artikel penelitian, dan publikasi lain yang membahas metode deteksi dan klasifikasi citra menggunakan teknik *deep learning*.

Dalam penelitian ini, digunakan metode YOLO (*You Only Look Once*) pada tahap deteksi karena kecepatan dan efisiensinya dalam mengenali objek, termasuk objek kecil seperti retakan tipis di jalan. Setelah area retakan terdeteksi dan dipotong menggunakan *bounding box*, tahap berikutnya adalah mengklasifikasi tingkat keparahan retakan tersebut. Untuk proses arsitektur ResNet50, yang terkenal handal dalam mengenali fitur visual yang rumit serta memiliki kinerja yang stabil berkat struktur residualnya.

Dari hasil telaah terhadap berbagai penelitian sebelumnya, diperoleh gambaran yang jelas tentang bagaimana kombinasi antara YOLO dan ResNet50 bisa dimanfaatkan secara efektif untuk membangun sistem pendeteksi dan klasifikasi retakan jalan berbasis citra digital

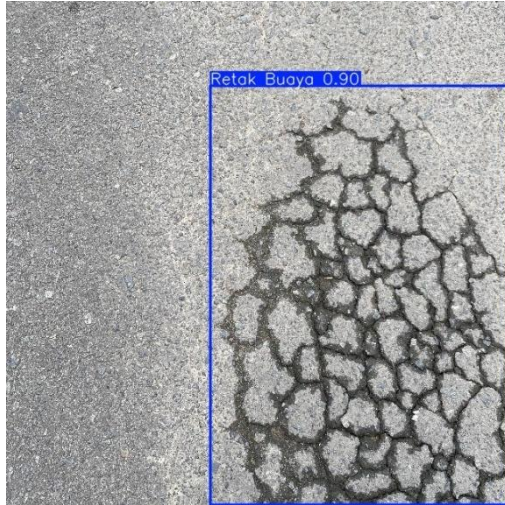
3.1.2 Pengumpulan Dataset

Dalam penelitian ini, pengumpulan dataset dilakukan dengan menggabungkan dua sumber utama. Dataset publik yang tersedia secara online seperti Roboflow digunakan sebagai data latih. Dataset ini menyediakan variasi yang cukup untuk melatih model YOLO dan CNN agar mampu mengenali berbagai jenis kerusakan secara akurat. Sedangkan untuk data uji, saya melakukan pengambilan gambar secara langsung di lapangan menggunakan kamera, sehingga model dapat diuji performanya pada kondisi nyata yang representatif. Pendekatan ini memungkinkan pelatihan model dengan data yang beragam sekaligus pengujian yang sesuai dengan kondisi lapangan sesungguhnya.

Pembagian dataset merupakan salah satu aspek penting dalam memastikan model dapat mempelajari data secara efektif dan bekerja dengan baik pada kondisi nyata adalah pembagian dataset. Dataset biasanya dipisahkan ke dalam tiga bagian, yaitu *training*, *validation*, dan *testing*. Pembagian ini memungkinkan model untuk mengenali berbagai variasi data sehingga dapat menghasilkan prediksi yang lebih akurat. Beberapa metode umum dalam membagi dataset antara lain 90:10, 80:20, atau 70:20:10. Dalam penelitian ini, dataset dibagi menjadi 70% untuk data latih 20% untuk data validasi dan 10% untuk data uji.

3.1.3. Anotasi Data

Dalam penelitian ini, setiap citra jalan dianotasi menggunakan perangkat lunak Roboflow dengan cara menandai bagian yang mengalami retakan menggunakan *bounding box*. Setiap retakan yang ditandai kemudian diberi label retak buaya.



Gambar 10. Contoh *Bounding Box Object Yolo*

3.1.4. *Preprocessing*

Preprocessing data merupakan tahap awal yang dilakukan untuk menyiapkan citra sebelum digunakan dalam proses pelatihan dan pengujian model. Tahap ini dilakukan untuk menyesuaikan karakteristik citra dengan kebutuhan masing-masing model, yaitu model deteksi YOLO dan model klasifikasi CNN ResNet50. Dengan adanya tahapan *preprocessing*, citra yang digunakan menjadi lebih seragam dan sesuai dengan format yang diharapkan oleh masing-masing model, sehingga proses pelatihan dapat berjalan secara optimal.

Pada tahap *preprocessing* untuk YOLO, citra jalan dilakukan proses *resize* agar sesuai dengan ukuran *input* yang diperlukan oleh model deteksi. Variasi ukuran citra yang digunakan dalam pengujian meliputi 320×320, 416×416, dan 640×640 piksel.

Penyesuaian ukuran citra ini bertujuan untuk menyeragamkan dimensi citra agar dapat diproses oleh arsitektur YOLO dengan baik. Selain itu, pengujian pada beberapa ukuran berbeda juga dilakukan untuk menemukan keseimbangan antara kecepatan deteksi dan tingkat akurasi model. Dengan demikian, proses

deteksi dapat dilakukan secara efisien tanpa mengurangi ketepatan identifikasi kerusakan jalan.

Citra hasil deteksi YOLO kemudian digunakan sebagai *Region of Interest (ROI)* untuk tahap klasifikasi tingkat kerusakan menggunakan CNN ResNet50. Pada tahap ini, dilakukan proses *cropping* untuk memotong bagian citra sesuai dengan area yang terdeteksi sebagai kerusakan berdasarkan hasil bounding box YOLO.

Setelah proses *cropping*, citra ROI kemudian dilakukan *resize* agar sesuai dengan ukuran input model CNN. Ukuran yang digunakan dalam pengujian meliputi 128×128, 224×224, dan 256×256 piksel.

Tahapan ini bertujuan untuk memastikan bahwa model CNN menerima citra yang berfokus pada area kerusakan jalan saja, sehingga proses klasifikasi dapat dilakukan secara lebih spesifik dan akurat. Dengan demikian, CNN dapat mempelajari karakteristik visual dari tingkat kerusakan retak buaya secara lebih efektif.

3.1.5. Pelatihan Model Yolo

Tahap pelatihan model YOLO menjadi inti dari penelitian ini, di mana model YOLOv12 dilatih menggunakan dataset yang telah melewati tahap *preprocessing* dan anotasi. Pada tahap ini, gambar yang telah diberi label dan *bounding box* dimasukkan ke dalam model, sehingga YOLO dapat mempelajari ciri-ciri visual dari berbagai jenis kerusakan jalan aspal. Proses pelatihan dilakukan dengan mengatur parameter-parameter agar diperoleh performa deteksi yang optimal. Hasil dari tahap ini adalah model YOLOv12 yang telah terlatih dan siap diimplementasikan untuk mendeteksi kerusakan jalan aspal secara otomatis pada citra baru.

3.1.6. Klasifikasi Tingkat Kerusakan Dengan CNN

Pada tahap klasifikasi tingkat keparahan kerusakan jalan aspal, gambar yang telah di-*crop* dari hasil deteksi model YOLO digunakan sebagai *input* bagi jaringan Convolutional Neural Network (CNN) dengan arsitektur ResNet-50. Arsitektur ini mencakup beberapa jenis lapisan utama, seperti lapisan konvolusi, lapisan *aktivasi*, *pooling*, *batch normalization*, serta *fully connected layer*. Setiap komponen ini berperan dalam mengekstraksi ciri visual penting dari citra, seperti tekstur retakan, ketebalan kerusakan, dan pola distribusinya, yang digunakan untuk mengklasifikasikan tingkat keparahan kerusakan menjadi tiga kategori: ringan, sedang, dan berat.

Salah satu elemen penting dalam CNN adalah fungsi aktivasi, yang bertugas menambahkan sifat non-linear pada jaringan, sehingga jaringan dapat mempelajari pola visual yang kompleks dari data. Dalam penelitian ini, fungsi aktivasi utama yang digunakan adalah ReLU (*Rectified Linear Unit*), yang berfungsi dengan menekan nilai negatif menjadi nol dan mempertahankan nilai positif tanpa

perubahan. Pendekatan ini mempercepat proses pembelajaran sekaligus mengurangi risiko masalah *vanishing gradient*, terutama dalam jaringan yang memiliki banyak lapisan.

Selain ReLU, penelitian ini juga mengkaji dua fungsi aktivasi alternatif, yaitu Leaky ReLU dan Swish. Leaky ReLU memungkinkan sebagian kecil nilai negatif tetap mengalir, sehingga mencegah aktivasi neuron menjadi mati secara permanen (dying ReLU). Sementara itu, Swish adalah fungsi aktivasi yang dikembangkan oleh Google dan memiliki sifat lebih halus serta adaptif. Fungsi ini dirumuskan sebagai $f(x) = x * \text{sigmoid}(x)$ dan terbukti efektif dalam meningkatkan kinerja model pada klasifikasi berbasis citra kompleks, termasuk dalam kasus kerusakan jalan.

Pada struktur ResNet-50, digunakan residual block, yaitu blok jaringan yang memiliki koneksi pintas (*shortcut connections*), memungkinkan sinyal *input* diteruskan langsung ke *output* tanpa harus melalui seluruh lapisan di dalam blok. Hal ini menjaga kestabilan propagasi gradien dan memungkinkan pelatihan jaringan yang sangat dalam tanpa menurunnya akurasi (*degradation problem*). Setelah fitur diekstraksi dari semua blok residual, data diteruskan ke lapisan *fully connected* yang menggunakan fungsi aktivasi Softmax. Fungsi ini mengubah *output* menjadi nilai probabilitas untuk setiap kelas kerusakan.

Model dilatih menggunakan variasi jumlah epoch sebanyak 30, 40, dan 50, guna mengamati pengaruh jumlah iterasi terhadap hasil klasifikasi. Eksperimen ini bertujuan untuk menentukan konfigurasi optimal yang menghasilkan keseimbangan antara akurasi pelatihan dan validasi, serta mencegah terjadinya *overfitting*. Pemilihan model terbaik didasarkan pada performa terhadap data validasi, yang mencerminkan kemampuan model dalam melakukan klasifikasi pada data baru yang belum pernah dilihat sebelumnya.

Secara keseluruhan, kombinasi antara ekstraksi fitur melalui layer konvolusi, penggunaan berbagai fungsi aktivasi non-linear seperti ReLU, Leaky ReLU, dan Swish, serta klasifikasi akhir menggunakan Softmax, memungkinkan CNN mengenali dan mengelompokkan tingkat kerusakan jalan aspal dengan lebih tepat berdasarkan informasi visual yang terkandung dalam citra.

3.1.7. Evaluasi Model

Evaluasi dilakukan menggunakan beberapa metrik umum, seperti *accuracy*, *precision*, *recall*, dan *mean Average Precision (mAP)*. Metrik-metrik ini diperoleh dari confusion matrix, yang pada dasarnya membandingkan hasil prediksi model dengan data aslinya untuk melihat seberapa tepat model dalam mengenali objek.

Berikut adalah penjelasan dari masing-masing metrik evaluasi: Akurasi digunakan untuk mengetahui seberapa sering model memberikan prediksi yang benar secara keseluruhan. Rumus untuk menghitung akurasi adalah:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Penjelasan:

1. TP (*True Positive*): Ketika model memprediksi hasil positif dan ternyata prediksinya benar.
2. TN (*True Negative*): Ketika model memprediksi hasil negatif dan prediksinya sesuai dengan kondisi sebenarnya.
3. FP (*False Positive*): Saat model memberikan prediksi positif, tapi kenyataannya salah atau tidak sesuai.
4. FN (*False Negative*): Ketika model memprediksi negatif, padahal sebenarnya hasilnya positif.

Presisi digunakan untuk menghitung seberapa tepat model saat memprediksi data sebagai positif. Dengan kata lain, dari semua hasil yang diprediksi positif oleh model, berapa banyak yang memang benar-benar positif.

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

Recall digunakan untuk mengukur seberapa banyak data positif yang berhasil dikenali oleh model. Jadi, dari seluruh data yang seharusnya dikategorikan sebagai positif, seberapa besar bagian yang berhasil diprediksi dengan benar oleh model.

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

F1-score merupakan rata-rata harmonis dari precision dan *recall* yang berfungsi untuk menyeimbangkan kedua metrik tersebut. Metrik ini sangat bermanfaat saat menghadapi dataset yang tidak seimbang, di mana precision dan *recall* sama-sama penting untuk diperhatikan.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

3.1.8. Kesimpulan

Penelitian ini memadukan metode YOLO untuk mendeteksi kerusakan pada jalan aspal secara otomatis dengan CNN berbasis arsitektur ResNet-50 yang berfungsi mengklasifikasikan tingkat keparahan kerusakan berdasarkan gambar hasil deteksi. Proses penelitian dimulai dengan kajian pustaka untuk memahami jenis-jenis kerusakan dan metode deteksi citra, kemudian dilanjutkan dengan

pengumpulan dan anotasi data, serta tahap *preprocessing* gambar. Selanjutnya, model YOLO dilatih untuk mendeteksi lokasi kerusakan, lalu hasil deteksi tersebut diklasifikasikan menggunakan ResNet-50 menjadi kategori kerusakan ringan, sedang, dan berat. Kinerja model dievaluasi menggunakan metrik seperti akurasi, *precision*, *recall*, dan *F1-score* agar hasilnya dapat dipercaya dan memiliki kemampuan generalisasi yang baik. Dengan cara ini, penelitian diharapkan bisa memberikan solusi sistem deteksi dan klasifikasi kerusakan jalan yang lebih cepat, tepat, dan membantu dalam perawatan jalan yang lebih efisien.

3.2. Alat dan Bahan.

Tabel 2. Estimasi biaya

No.	Alat/bahan	Harga Satuan (Rp.)	Jumlah	Total (Rp.)	Keterangan ¹
1.	Laptop Lenovo Ideapad Gaming 3	-	1	-	Milik Pribadi
2.	Smart Phone	-	1	-	Milik Pribadi
3.	Webcam	-	1	-	Milik Pribadi
Total				-	-

3.3. Pengujian ukuran citra

Pengujian *preprocessing* dilakukan dengan melakukan proses *resizing* citra input agar sesuai dengan kebutuhan arsitektur model YOLO dan CNN ResNet50. Variasi ukuran citra yang digunakan dalam pengujian ini meliputi ukuran yang umum diterapkan pada model YOLO, yaitu 320×320, 416×416, dan 640×640 piksel, serta 128×128, 224×224, dan 256×256 piksel untuk CNN ResNet-50. Pengujian variasi ukuran *input* ini dilakukan untuk mengetahui pengaruh resolusi citra terhadap performa model YOLO dan CNN ResNet50 .[28]

Tabel 3. Pengujian Ukuran Citra YOLO

Metode	Ukuran Citra	MaP50	<i>Precision%</i>	<i>Recall%</i>
YOLO (deteksi)	320 x 320			
	416 x 416			
	640 x 640			

Tabel 4. Pengujian Ukuran Citra CNN

Metode	Ukuran Citra	Accuracy%	Precision%	Recall%	F1-Score%
CNN (Klasifikasi)	128 x 128				
	224 x 224				
	256 x 256				

3.4. Pengujian CNN Dengan Fungsi Aktivasi Yang Berbeda.

Setelah melakukan analisis terhadap berbagai variasi ukuran citra pada tahap *preprocessing*, ditemukan bahwa ukuran citra dengan nilai akurasi tertinggi dipilih sebagai ukuran citra optimal untuk digunakan dalam seluruh proses pengujian sistem berikutnya. Oleh karena itu, pengujian sistem difokuskan pada model CNN yang telah menggunakan ukuran citra terbaik tersebut, tanpa melakukan variasi ulang pada *preprocessing* ukuran citra

Pada tahap ini, dilakukan evaluasi terhadap beberapa jenis fungsi aktivasi yang diterapkan pada arsitektur CNN, yakni ReLU, Leaky ReLU. Tujuan dari pengujian ini adalah untuk menganalisis bagaimana masing-masing fungsi aktivasi memengaruhi kinerja dalam mengklasifikasikan tingkat kerusakan jalan aspal. Fungsi aktivasi memiliki peran krusial dalam membantu jaringan saraf mengenali dan mempelajari pola dari data *input*.

Pengujian ini menggunakan arsitektur CNN ResNet-50 dengan optimizer Adam dan jumlah epoch yang ditetapkan yaitu 30. Dan ada 3 kelas, yaitu ringan, sedang, dan berat.

Tabel 5. Pengujian Fungsi Aktivasi CNN

Fungsi Aktivasi	Accuracy%	Precision%	Recall%	F1- Score%
ReLU				
Leaky ReLU				
Swish				

3.5. Pengujian CNN Klasifikasi Dengan Jumlah Epoch yang Berbeda.

Setelah pengujian terhadap berbagai fungsi aktivasi selesai dilakukan, tahap selanjutnya adalah menguji pengaruh jumlah epoch yang berbeda. Tujuan dari pengujian ini adalah untuk mengetahui jumlah epoch yang paling efektif dalam melatih model CNN, sehingga model dapat mencapai akurasi terbaik tanpa mengalami *overfitting*.

Dalam pengujian ini digunakan arsitektur CNN ResNet-50, dengan optimizer Adam dan fungsi aktivasi yang sebelumnya menunjukkan performa terbaik. Variasi jumlah epoch yang diuji meliputi ,30, 40, dan 50

Tabel 6. Pengujian dengan Jumlah Epoch yang Berbeda

Jumlah Epoch	Accuracy%	Precision%	Recall%	F1- Score%
30				
40				
50				

3.6. Pengujian Dengan Learning Rate Yang Berbeda.

Untuk mengetahui pengaruh nilai *learning rate* terhadap kinerja model, dilakukan pengujian dengan tiga variasi *learning rate* , yaitu 0,001, 0,0001, 0,00001. Tujuan pengujian ini adalah untuk memperoleh nilai *learning rate* yang paling optimal sehingga model dapat bealajar stabil dan menghasilkan akurasi terbaik pada data uji.

Tabel 7. Pengujian Dengan Learning Rate

Learning Rate	Accuracy%	Precision%	Recall%	F1- Score%
0,001				
0,0001				
0,00001				

3.7. Pengujian Dengan Data Tambahan

Selain menggunakan data uji dari dataset utama, penelitian ini juga melakukan pengujian tambahan dengan menggunakan data yang tidak termasuk dalam data latih maupun data uji sebelumnya. Data tambahan ini bersifat *unseen*, artinya data yang tidak termasuk kedalam data latih maupun data uji sebelumnya. Tujuan dari pengujian ini adalah untuk mengetahui seberapa baik model CNN yang telah dilatih dapat menggeneralisasi atau menyesuaikan diri saat diberikan gambar baru yang berasal dari kondisi atau distribusi yang berbeda dari data pelatihan untuk mengklasifikasikannya.

Pada pengujian ini, beberapa gambar kerusakan jalan aspal diuji menggunakan model klasifikasi yang sudah jadi. Dari setiap proses prediksi, dicatat beberapa informasi penting seperti nama file gambar, label asli (kelas sebenarnya), hasil prediksi model, apakah prediksi tersebut benar atau salah.

Tabel 8. Pengujian Dengan Data Tambahan

No	Gambar	Kelas Sebenarnya	Prediksi Model	<i>Confidence Score%</i>
1				
2				
3				
4				
5				

3.8. Analisis Pengujian

Tabel 9. Analisis.

No	Jenis Pengujian	Tujuan Pengujian	Parameter yang Diamati	Kesimpulan
1.	Ukuran citra (YOLO&CNN)	Menentukan ukuran citra optimal untuk deteksi dan klasifikasi	Akurasi deteksi YOLO dan klasifikasi CNN	
2.	Fungsi Aktivasi CNN	Mengetahui pengaruh fungsi aktivasi terhadap performa klasifikasi	Akurasi, <i>Precision</i> , <i>Recall</i> , <i>F1-Score</i>	

3.	Jumlah Epoch CNN	Menganalisis efek jumlah epoch terhadap proses pelatihan dan <i>overfitting</i>	Akurasi, grafik <i>training & validation loss</i>	
5.	Learning Rate	Mengetahui pengaruh perbedaan nilai <i>learning rate</i> terhadap proses pembelajaran model CNN	Akurasi, grafik <i>training & validation loss</i>	
4.	Data Tambahan	Menguji kemampuan generalisasi model terhadap data baru	Prediksi model	

Bab 4. Hasil dan Pembahasan

4.1 Pengambilan Data

Pengambilan gambar untuk pelatihan model dilakukan dengan memfokuskan sudut pengambilan yang mampu memperjelas pola serta tingkat keparahan kerusakan retak buaya pada permukaan jalan. Proses pengambilan data dilakukan pada beberapa ruas jalan, yaitu Jalan Palm Spring , Taman Baloi, Dan Jalan Brigjen Katamso-Komp,Sungai Binti. Seluruh pengambilan gambar dilakukan dalam kondisi pencahayaan yang baik pada siang hari untuk menjaga kejelasan tekstur permukaan jalan, sehingga model dapat mengekstraksi fitur visual secara optimal selama proses pelatihan maupun inferensi.

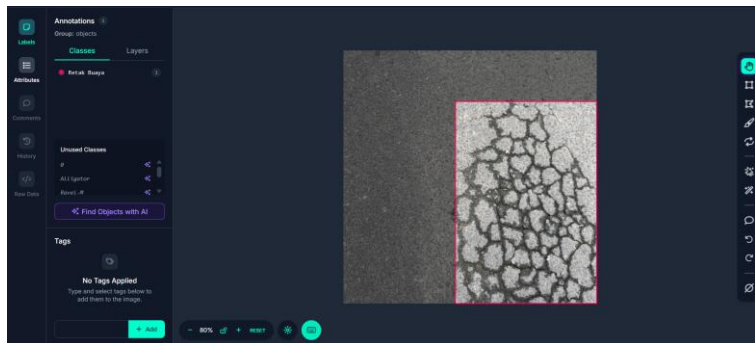
Sebanyak 1.156 gambar dikumpulkan dan digunakan sebagai dataset utama dalam penelitian ini. Seluruh citra disimpan dalam format JPG dan selanjutnya diproses serta dikelola menggunakan *platform* Roboflow untuk keperluan anotasi dan pembagian dataset.

Gambar 10 menampilkan contoh citra permukaan jalan dengan kerusakan retak buaya yang digunakan sebagai dataset penelitian. Citra tersebut merupakan hasil pengambilan langsung di lapangan dengan sudut pandang yang memperjelas pola retakan serta tingkat keparahan kerusakan, yang mencakup variasi kelas kerusakan ringan, sedang, dan berat. Kondisi pencahayaan yang memadai pada citra ini diharapkan dapat mendukung kinerja model dalam proses deteksi dan klasifikasi kerusakan jalan.



Gambar 10. Contoh Pengambilan Gambar

4.2 Anotasi Data



Gambar 11. Anotasi Menggunakan Roboflow

Pada gambar 11, setelah pengambilan gambar selesai, langkah berikutnya adalah mengunggah semua gambar kedalam platform yang di gunakan yaitu Roboflow, dataset akan di *anotasi* untuk menandai area kerusakan yang akan dideteksi oleh model. *Anotasi* ini bertujuan untuk memberikan label pada area tertentu dalam gambar agar model dapat mempelajari kerusakan jenis retak buaya dengan akurasi yang tinggi.

4.3 Preprocessing Citra

Preprocessing citra dilakukan untuk menyiapkan data sebelum digunakan pada dua tahap pemodelan, yaitu deteksi retak buaya menggunakan YOLO dan klasifikasi tingkat kerusakan menggunakan Convolutional Neural Network (CNN). Tahapan ini bertujuan untuk meningkatkan kualitas data, menyeragamkan karakteristik citra, serta memastikan kesesuaian data dengan kebutuhan masing-masing model.

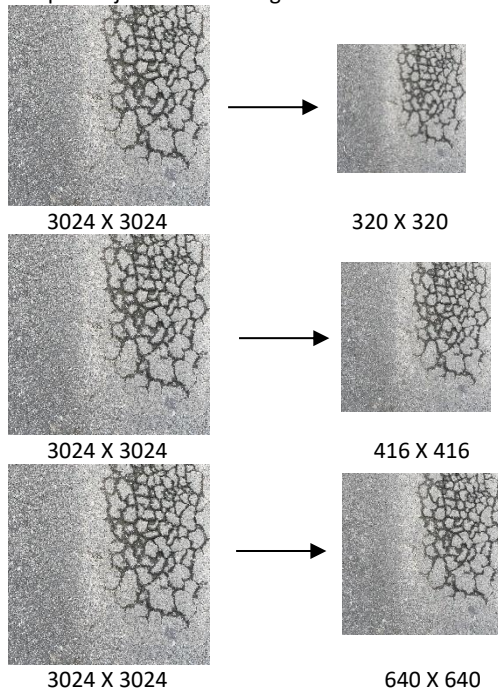
4.3.1. *Preprocessing* Citra Untuk YOLO

Pada tahap awal, *preprocessing* diterapkan pada seluruh dataset citra untuk keperluan pelatihan dan pengujian model YOLO. Mengingat data citra diperoleh dari kondisi lapangan dengan variasi ukuran, pencahayaan, dan tekstur permukaan jalan, diperlukan proses penyesuaian agar model dapat mempelajari pola retakan secara optimal.

4.3.2. *Resize image* YOLO

Pada tahap ini dilakukan proses *resize* atau penyeragaman ukuran citra untuk menyesuaikan seluruh dataset dengan format *input* yang dibutuhkan oleh model YOLO. Citra hasil pengambilan lapangan memiliki ukuran bervariasi, sehingga diperlukan proses perubahan ukuran agar model dapat melakukan proses pelatihan dan inferensi secara konsisten.

Proses *resize* dilakukan terhadap seluruh citra pada dataset dengan tiga variasi ukuran, yaitu 320×320 piksel, 416×416 piksel, dan 640×640 piksel. Variasi ini digunakan sebagai parameter pengujian untuk mengetahui pengaruh ukuran citra terhadap kinerja model dari segi akurasi.



Gambar 12 Penerapan Ukuran Citra Model YOLO

Pada Gambar 12 menunjukkan bahwa setiap citra telah berhasil diseragamkan menjadi bentuk persegi dengan proporsi retakan tetap terjaga. Proses ini juga memastikan setiap citra memiliki resolusi yang sesuai dengan spesifikasi arsitektur YOLO. Hasil dari tahap ini kemudian digunakan sebagai data *input* pada proses pelatihan dan pengujian model.

4.3.3 Preprocessing Citra Untuk CNN

Setelah tahap deteksi retak buaya menggunakan model YOLO, dilakukan *preprocessing* lanjutan terhadap citra hasil deteksi sebelum digunakan sebagai data masukan pada model Convolutional Neural Network (CNN). Tahapan *preprocessing* ini bertujuan untuk memastikan bahwa CNN hanya menerima informasi visual yang relevan dengan area retakan, serta memiliki kesesuaian ukuran dan kualitas data untuk proses pelatihan dan pengujian.

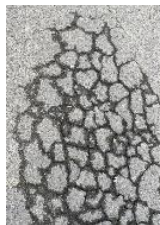
Preprocessing citra pada tahap ini difokuskan pada ekstraksi area retakan dan penyesuaian karakteristik citra sesuai dengan kebutuhan CNN. Dengan demikian, CNN dapat lebih optimal dalam mempelajari perbedaan tekstur dan tingkat kerapatan retak yang menjadi dasar klasifikasi tingkat kerusakan ringan, sedang, dan berat.

4.3.4 Cropping Citra

Cropping citra dilakukan berdasarkan *bounding box* hasil deteksi model YOLO terhadap area retak buaya. Proses ini bertujuan untuk mengekstraksi area retakan secara spesifik sehingga bagian citra yang tidak relevan, seperti permukaan jalan tanpa kerusakan atau objek lain di sekitar area retak, dapat dieliminasi.

Dengan mengisolasi area retakan, citra hasil *cropping* menjadi lebih fokus pada pola visual retak buaya. Hal ini membantu CNN dalam mengekstraksi fitur yang lebih representatif terkait tekstur, kerapatan, dan tingkat kerusakan retakan. Selain itu, proses *cropping* juga berkontribusi dalam mengurangi gangguan latar belakang yang dapat mempengaruhi akurasi klasifikasi.

Pada Gambar 13 merupakan citra hasil pemotongan (*cropping*) berdasarkan koordinat *bounding box* yang dihasilkan oleh model YOLO. Citra ini merepresentasikan data input spesifik yang akan digunakan untuk pelatihan dan pengujian model klasifikasi CNN. Seperti yang terlihat, setiap sub-citra hasil *cropping* telah berhasil mengisolasi objek retakan (atau area non-retakan) dan menghilangkan sebagian besar latar belakang jalan, sehingga model klasifikasi dapat lebih fokus dalam mempelajari fitur internal retakan tanpa terganggu oleh detail lingkungan yang tidak relevan.



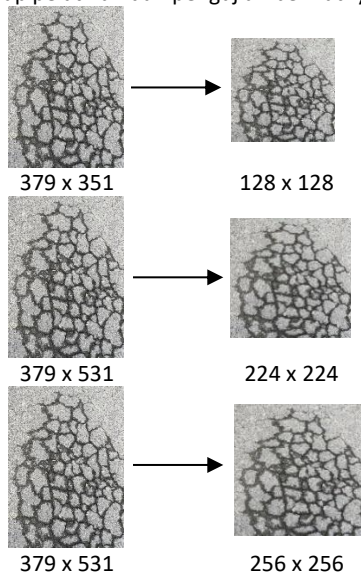
Gambar 13 Citra Hasil *Cropping*

4.3.5 *Resize image CNN*

Pada tahap ini dilakukan proses *resize* terhadap dataset citra yang akan digunakan untuk pelatihan model CNN. Proses ini bertujuan untuk menyeragamkan ukuran setiap citra agar sesuai dengan format *input* yang dibutuhkan oleh arsitektur CNN yang digunakan, yaitu ResNet50.

Setiap citra hasil *cropping* dari deteksi YOLO disesuaikan ukurannya menjadi 224×224 piksel, 128×128 piksel, dan 256×256 piksel. Variasi ukuran ini digunakan untuk mengetahui pengaruh resolusi input terhadap kemampuan model dalam mengklasifikasikan tingkat keparahan kerusakan retak buaya.

Proses perubahan ukuran dilakukan menggunakan untuk mempertahankan detail tekstur pada citra. Setelah dilakukan *resize*, seluruh citra tersimpan dengan format dan dimensi yang seragam sehingga dapat diproses oleh model CNN pada tahap pelatihan dan pengujian berikutnya.



Gambar 14 Penerapan Ukuran Citra Model CNN

Pada Gambar 14 hasil dari tahap *resize* menunjukkan bahwa bentuk dan pola retakan tetap terjaga meskipun ukuran citra berubah. Tahapan ini memastikan konsistensi ukuran input sehingga model CNN dapat melakukan proses ekstraksi fitur dengan optimal.

4.4. Pengujian Model Deteksi YOLO

Pengujian model deteksi objek YOLO dilakukan untuk mengevaluasi kemampuan model dalam mendeteksi area kerusakan jalan aspal pada citra. Pada penelitian ini digunakan model YOLOv12n (YOLO versi 12 varian nano) yang dikenal memiliki arsitektur ringan dan efisien, sehingga sesuai untuk penerapan pada sistem *real-time*. Model YOLOv12n dilatih menggunakan dataset citra kerusakan jalan yang telah melalui tahap *preprocessing*.

Evaluasi kinerja model dilakukan menggunakan metrik standar pada deteksi objek, yaitu *Mean Average Precision* (mAP@0.5), *Precision*, dan *Recall*, untuk mengukur tingkat ketepatan, kelengkapan deteksi, serta kualitas keseluruhan hasil deteksi.

4.4.1. Hasil Pengujian Variasi Ukuran Citra Pada Model YOLO

Pengujian variasi ukuran citra pada model YOLOv12n dilakukan untuk mengetahui pengaruh resolusi citra terhadap kinerja deteksi objek kerusakan jalan. Variasi ukuran citra yang digunakan pada pengujian ini yaitu 320×320, 416×416, dan 640×640. Parameter lain seperti arsitektur model, *learning rate*, *batch size*, dan jumlah epoch dijaga tetap sama agar performa yang dihasilkan hanya dipengaruhi oleh perubahan ukuran citra.

Tabel 10. Hasil Pengujian Ukuran Citra

Metode	Ukuran Citra	MaP50	<i>Precision</i> %	<i>Recall</i> %
YOLO (deteksi)	320 x 320	90.02%	92%	81%
	416 x 416	90.08%	91.02%	81.04%
	640 x 640	90.08%	92.08%	83.02%

Berdasarkan Tabel 10, variasi ukuran citra memberikan pengaruh terhadap performa deteksi model YOLOv12n, meskipun perbedaannya tidak terlalu besar. Pada ukuran 320 × 320, model memperoleh nilai mAP50 sebesar 90.02% dengan *precision* 92%. Namun, nilai *recall* masih berada pada 81%, yang menunjukkan bahwa sebagian objek kerusakan masih terlewat. Hal ini wajar karena citra beresolusi kecil memiliki detail fitur yang terbatas sehingga informasi penting pada area kerusakan tidak sepenuhnya tertangkap.

Pada ukuran 416 × 416, performa model menjadi lebih stabil. Nilai mAP50 meningkat sedikit menjadi 90.08%, *precision* berada di angka 91.02%, dan *recall* naik menjadi 81.04%. Meskipun peningkatan *recall* tidak terlalu besar, ukuran ini sudah mampu memberikan detail yang lebih baik dibandingkan 320×320.

Ukuran 640×640 menghasilkan performa terbaik dibandingkan ukuran lainnya. Nilai mAP50 tetap sebesar 90.08%, namun *precision* meningkat menjadi 92.08%, dan *recall* mencapai nilai tertinggi yaitu 83.02%. Peningkatan ini menunjukkan bahwa model mampu mengenali lebih banyak area kerusakan berkat detail citra yang lebih jelas dan lengkap. Dengan informasi visual yang lebih kaya, model memiliki peluang lebih besar untuk mendeteksi objek yang berukuran kecil atau memiliki tepi yang kurang jelas pada citra resolusi rendah.

Berdasarkan hasil tersebut, ukuran 640×640 dipilih sebagai ukuran citra yang digunakan pada implementasi *real-time*. Meskipun ukuran ini membutuhkan komputasi yang sedikit lebih besar, hasil deteksi yang lebih akurat sangat mendukung kebutuhan sistem, terutama dalam konteks klasifikasi dan integrasi dengan model CNN. Ukuran ini memberikan kombinasi yang optimal antara akurasi deteksi dan kestabilan prediksi pada berbagai kondisi citra.

4.5. Pengujian Model Klasifikasi CNN

Pada tahap ini, dilakukan serangkaian pengujian optimasi parameter pada model klasifikasi Convolutional Neural Network (CNN) dengan arsitektur ResNet-50. Pengujian ini bertujuan untuk menentukan konfigurasi *hyperparameter* optimal yang mampu memaksimalkan kinerja klasifikasi tingkat kerusakan jalan (Ringan, Sedang, Berat). Pengujian dilakukan secara berurutan, di mana hasil optimal dari satu parameter (misalnya, Ukuran Citra) akan digunakan sebagai *setting* dasar untuk pengujian parameter berikutnya (Fungsi Aktivasi, *Epoch*, dan *Learning Rate*).

4.5.1. Hasil Pengujian Variasi Ukuran Citra Pada Model CNN

Pengujian variasi ukuran citra pada model CNN dilakukan untuk mengetahui pengaruh resolusi citra terhadap kinerja klasifikasi tingkat kerusakan jalan. Pada pengujian ini digunakan tiga variasi ukuran citra, yaitu 128×128 , 224×224 , dan 256×256 . Parameter lain seperti arsitektur model, fungsi aktivasi, *learning rate*, batch size, dan jumlah epoch dijaga tetap sama agar hasil yang diperoleh mencerminkan pengaruh dari perubahan ukuran citra.

Tabel 11. Hasil Pengujian Ukuran Citra CNN

Metode	Ukuran Citra	<i>Accuracy%</i>	<i>Precision%</i>	<i>Recall%</i>	<i>F1-Score%</i>
CNN (Klasifikasi)	128 x 128	88.57%	90.29%	88.57%	88.75%
	224 x 224	92.38%	93.02%	92.38%	92.42%
	256 x 256	85.71%	87.52%	85.71%	85.89%

Berdasarkan pada Tabel 11, dapat diketahui bahwa variasi ukuran citra memberikan pengaruh yang signifikan terhadap kinerja klasifikasi model CNN. Pada ukuran citra 128x128, model menghasilkan nilai akurasi sebesar 88.57%, dengan nilai *precision*, *recall*, dan *F1-score* yang relatif baik. Namun, ukuran citra yang lebih kecil menyebabkan informasi fitur visual tidak tertangkap secara optimal, terutama untuk detail kerusakan jalan yang berukuran kecil atau memiliki tekstur yang kompleks.

Pada ukuran citra 224x224, performa model meningkat secara signifikan dengan nilai akurasi mencapai 92.38%, serta nilai *precision*, *recall*, dan *F1-score* yang juga paling tinggi dibandingkan ukuran citra lainnya. Hal ini menunjukkan bahwa ukuran citra 224x224 mampu memberikan keseimbangan antara kelengkapan informasi fitur dan efisiensi komputasi, sehingga model CNN dapat mempelajari pola karakteristik kerusakan jalan dengan lebih optimal.

Sementara itu, pada ukuran citra 256x256, performa model justru mengalami penurunan dengan nilai akurasi sebesar 85.71%. Penurunan ini mengindikasikan bahwa peningkatan resolusi citra secara berlebihan tidak selalu berdampak positif terhadap kinerja model. Ukuran citra yang terlalu besar dapat meningkatkan kompleksitas fitur yang dipelajari serta berpotensi menyebabkan kesulitan dalam proses optimasi model, terutama karena jumlah data latih dari dataset pada penelitian ini terbatas.

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, ukuran citra 224x224 ditetapkan sebagai ukuran citra optimal untuk model CNN. Pemilihan ini didasarkan pada nilai *accuracy*, *precision*, *recall*, dan *F1-score* tertinggi, serta keseimbangan antara performa klasifikasi dan efisiensi komputasi. Ukuran citra ini selanjutnya digunakan pada pengujian parameter lainnya dan pada tahap evaluasi model CNN secara keseluruhan.

4.5.2. Hasil Pengujian Variasi Fungsi Aktivasi Pada Model CNN

Pengujian ini bertujuan untuk menganalisis pengaruh fungsi aktivasi yang berbeda, yaitu ReLU, Leaky ReLU, dan Swish, terhadap kinerja klasifikasi model CNN. Pengujian ini menggunakan ukuran citra optimal 224x224 yang telah ditetapkan pada sub-bab sebelumnya.

Tabel 12. Hasil Pengujian Fungsi Aktivasi

Fungsi Aktivasi	<i>Accuracy%</i>	<i>Precision%</i>	<i>Recall%</i>	<i>F1- Score%</i>
ReLU	92.38%	93.02%	92.38%	92.42%
Leaky ReLU	90.48%	91.31%	90.48%	90.43%
Swish	94.29%	94.59%	94.29%	94.26%

Berdasarkan hasil pengujian fungsi aktivasi yang disajikan pada Tabel 12, dapat dilihat bahwa jenis fungsi aktivasi memberikan pengaruh yang signifikan terhadap kinerja klasifikasi model CNN. Pengujian dilakukan dengan menggunakan ukuran citra optimal 224x224 piksel, sehingga perbedaan kinerja yang dihasilkan murni disebabkan oleh perbedaan fungsi aktivasi yang digunakan.

Fungsi aktivasi ReLU menghasilkan nilai akurasi sebesar 92,38% dengan nilai *precision* dan *recall* yang relatif tinggi. Hal ini menunjukkan bahwa ReLU masih mampu melakukan pembelajaran fitur dengan baik, namun memiliki keterbatasan dalam menangani neuron yang tidak aktif (*dead neuron*).

Pada fungsi aktivasi Leaky ReLU, terjadi penurunan nilai akurasi menjadi 90,48%. Meskipun Leaky ReLU dirancang untuk mengurangi permasalahan *dead neuron* dengan memberikan gradien kecil pada nilai negatif, hasil pengujian menunjukkan bahwa kinerjanya masih lebih rendah dibandingkan ReLU. Hal ini dapat disebabkan oleh karakteristik dataset yang tidak terlalu kompleks sehingga keuntungan Leaky ReLU tidak terlalu signifikan.

Fungsi aktivasi Swish menunjukkan performa terbaik dengan nilai akurasi sebesar 94,29%, serta nilai *precision*, *recall*, dan *F1-score* tertinggi dibandingkan fungsi aktivasi lainnya. Hal ini mengindikasikan bahwa Swish mampu menangani non-linearitas dengan lebih baik serta mempertahankan aliran gradien yang lebih stabil selama proses pelatihan, sehingga fitur-fitur penting pada citra kerusakan jalan dapat dipelajari secara lebih optimal.

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, fungsi aktivasi Swish dipilih sebagai fungsi aktivasi terbaik untuk digunakan pada model CNN ResNet-50, karena menghasilkan kinerja klasifikasi tertinggi dan paling stabil pada seluruh metrik evaluasi. Oleh karena itu, fungsi aktivasi Swish digunakan pada pengujian selanjutnya dalam penelitian ini.

4.5.3. Hasil Pengujian Variasi Jumlah Epoch

Pengujian variasi jumlah epoch dilakukan untuk menganalisis pengaruh jumlah iterasi pelatihan terhadap kinerja model CNN dalam mengklasifikasikan tingkat kerusakan jalan. Pengujian ini menggunakan arsitektur CNN ResNet-50, fungsi aktivasi Swish, optimizer Adam, ukuran citra 224×224 piksel, serta parameter lain yang dibuat tetap untuk menjaga konsistensi hasil pengujian.

Variasi jumlah epoch yang digunakan pada pengujian ini adalah 30 epoch, 40 epoch, dan 50 epoch. Hasil pengujian disajikan pada Tabel 13.

Tabel 13. Hasil Pengujian Variasi Jumlah Epoch

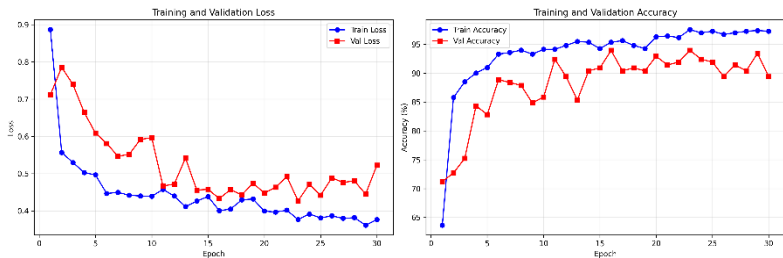
Jumlah Epoch	<i>Accuracy%</i>	<i>Precision%</i>	<i>Recall%</i>	<i>F1- Score%</i>
30	95.24%	95.30%	95.24%	95.18%
40	91.43%	92.72%	91.43%	91.33%
50	95.24%	95.41%	95.24%	95.23%

Berdasarkan Tabel 13, dapat diketahui bahwa variasi jumlah epoch memberikan pengaruh yang signifikan terhadap performa model CNN. Pada 30 epoch, model telah mampu mencapai nilai akurasi yang tinggi sebesar 95.24%, dengan nilai *precision*, *recall*, dan *F1-score* yang juga tinggi dan seimbang. Hal ini menunjukkan bahwa model sudah dapat mempelajari pola fitur citra kerusakan jalan dengan baik.

Pada 40 epoch, performa model justru mengalami penurunan, ditandai dengan menurunnya nilai akurasi menjadi 91.43%. Penurunan ini mengindikasikan bahwa penambahan epoch tidak selalu berbanding lurus dengan peningkatan performa model dan dapat menyebabkan ketidakstabilan proses pembelajaran.

Sementara itu, pada 50 epoch, nilai akurasi kembali meningkat menjadi 95.24%, dengan nilai *precision* dan *F1-score* yang sedikit lebih tinggi dibandingkan 30 epoch. Meskipun demikian, peningkatan performa pada 50 epoch tidak signifikan dibandingkan dengan 30 epoch, sehingga dari sisi efisiensi pelatihan, 30 epoch dinilai lebih optimal.

Analisis Grafik Loss dan Accuracy



Gambar 15 Grafik Training dan Validation Loss serta Accuracy pada Epoch 30

Pada Gambar 13 menunjukkan grafik *training learning curve* dan *validation learning curve* pada epoch 30. Berdasarkan grafik *training* dan *validation loss*, terlihat bahwa nilai *training loss* mengalami penurunan yang signifikan pada epoch-epoch awal dan terus menurun secara bertahap hingga mencapai kondisi relatif stabil pada epoch selanjutnya. *Validation loss* juga menunjukkan tren menurun dengan fluktuasi yang masih dalam batas wajar. Selisih antara *training loss* dan *validation loss* relatif kecil, yang menandakan bahwa model mampu melakukan generalisasi dengan baik serta tidak mengalami *overfitting* yang signifikan.

Pada grafik *training* dan *validation accuracy*, terlihat bahwa nilai *training accuracy* meningkat secara cepat pada epoch awal dan terus meningkat hingga mencapai nilai di atas 95%. *Validation accuracy* juga menunjukkan peningkatan yang sejalan dengan *training accuracy*, meskipun terdapat gejala naik turun kecil pada beberapa epoch. Namun, secara keseluruhan *validation accuracy* cenderung stabil pada epoch-epoch akhir, yang menandakan bahwa model telah mencapai kondisi stabil.

Berdasarkan hasil pengujian pada Tabel 13 serta analisis grafik *learning curve*, jumlah epoch optimal ditentukan pada 30 epoch. Pemilihan ini didasarkan pada kondisi di mana nilai *validation accuracy* telah mencapai nilai maksimum, *validation loss* cenderung stabil, serta waktu pelatihan yang lebih efisien dibandingkan dengan penggunaan epoch yang lebih besar.

4.5.3. Hasil Pengujian Variasi *Learning Rate*

Pengujian variasi learning rate dilakukan untuk mengetahui pengaruh besar langkah pembaruan bobot terhadap kinerja model CNN dalam mengklasifikasikan tingkat kerusakan jalan. Pada pengujian ini digunakan arsitektur CNN ResNet-50 dengan fungsi aktivasi Swish, optimizer Adam, ukuran citra 224×224 piksel, serta jumlah epoch 30 sebagai epoch optimal. Parameter lain seperti *batch size* dijaga tetap agar pengaruh yang diamati hanya berasal dari variasi nilai learning rate.

Variasi learning rate yang diuji terdiri dari 0,001, 0,0001, dan 0,00001. Hasil pengujian disajikan pada Tabel 14.

Tabel 14. Hasil Pengujian *Learning Rate*

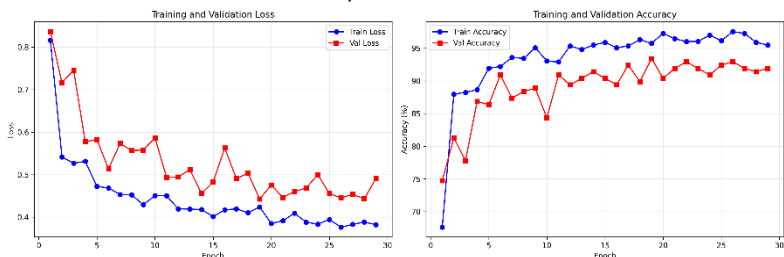
<i>Learning Rate</i>	<i>Accuracy%</i>	<i>Precision%</i>	<i>Recall%</i>	<i>F1- Score%</i>
0,001	85.71%	86.70%	85.71%	85.55%
0,0001	93.33%	94.01%	93.33%	93.30%
0,00001	91.43%	91.80%	91.43%	91.50%

Berdasarkan Tabel 14, dapat diketahui bahwa variasi nilai *learning rate* memberikan pengaruh yang signifikan terhadap performa model CNN. Pada *learning rate* 0,001, model menghasilkan nilai akurasi terendah, yaitu 85,71%, yang menunjukkan bahwa proses pembelajaran belum berjalan secara optimal. Nilai *learning rate* yang terlalu besar menyebabkan pembaruan bobot berlangsung secara agresif, sehingga model sulit menemukan titik konvergensi yang stabil.

Pada *learning rate* 0,0001, performa model meningkat secara signifikan dengan nilai akurasi sebesar 93,33%, serta diikuti oleh peningkatan nilai *precision*, *recall*, dan *F1-score* yang relatif seimbang. Hal ini menunjukkan bahwa nilai *learning rate* tersebut mampu memberikan keseimbangan antara kecepatan pembelajaran dan kestabilan optimasi.

Sementara itu, pada *learning rate* 0,00001, performa model sedikit menurun dibandingkan *learning rate* 0,0001, dengan nilai akurasi sebesar 91,43%. Penurunan ini mengindikasikan bahwa nilai *learning rate* yang terlalu kecil menyebabkan pembaruan bobot berlangsung sangat lambat, sehingga model belum mencapai performa optimal meskipun jumlah epoch yang digunakan sama.

Analisis Grafik Loss dan Accuracy



Gambar 16. Grafik Training dan Validation Loss Pada *Learning Rate* 0,0001

Grafik *training* dan *validation loss* serta *training* dan *validation accuracy* pada pengujian variasi *learning rate* 0,0001 ditunjukkan pada Gambar 14. Berdasarkan grafik *training* dan *validation loss*, terlihat bahwa nilai *training loss* mengalami

penurunan yang konsisten sejak epoch awal hingga akhir pelatihan. Sementara itu, *validation loss* juga menunjukkan tren penurunan dengan fluktuasi yang relatif kecil. Perbedaan antara *training loss* dan *validation loss* tidak terlalu besar, yang menandakan bahwa model mampu belajar dengan baik sekaligus mempertahankan kemampuan generalisasi terhadap data validasi.

Pada grafik *training* dan *validation accuracy*, terlihat bahwa nilai *training accuracy* meningkat secara signifikan pada epoch-epoch awal, kemudian meningkat secara bertahap hingga mencapai kondisi stabil. *Validation accuracy* menunjukkan pola peningkatan yang sejalan dengan *training accuracy*, meskipun terdapat fluktuasi kecil pada beberapa epoch. Fluktuasi tersebut masih dalam batas wajar dan tidak menunjukkan gejala *overfitting* yang signifikan.

Secara konseptual, nilai *learning rate* 0,0001 memungkinkan proses optimasi berjalan lebih stabil dibandingkan *learning rate* yang lebih besar (0,001) maupun yang lebih kecil (0,00001). Nilai ini memberikan pembaruan bobot yang cukup untuk mempercepat konvergensi, tanpa menyebabkan lonjakan nilai *loss* atau ketidakstabilan pelatihan.

Berdasarkan hasil pada Tabel 14 serta analisis grafik *learning curve*, *learning rate* optimal dalam penelitian ini ditentukan pada nilai 0,0001. Pemilihan ini didasarkan pada capaian nilai akurasi tertinggi, keseimbangan metrik evaluasi, serta pola *training* dan *validation loss* yang stabil. Nilai *learning rate* tersebut selanjutnya digunakan pada tahapan pengujian parameter lainnya serta pada proses integrasi sistem secara *real-time*.

4.5.4. Evaluasi Model Akhir CNN

Pada tahap ini evaluasi model CNN dilakukan setelah seluruh parameter optimal ditentukan melalui tahap pengujian sebelumnya. Parameter terbaik yang digunakan pada model CNN meliputi ukuran citra 224×224, fungsi aktivasi Swish, jumlah epoch 30, dan *learning rate* 0.0001. Evaluasi ini bertujuan untuk menilai kemampuan generalisasi model CNN dalam mengklasifikasikan tingkat kerusakan jalan aspal ke dalam tiga kelas, yaitu ringan, sedang, dan berat.

Evaluasi dilakukan menggunakan tiga subset data, yaitu data latih, data validasi, dan data uji. Metrik evaluasi yang digunakan meliputi akurasi, *loss*, *precision*, *recall*, dan *F1-score*. Hasil evaluasi kinerja model CNN ditunjukkan pada Tabel 15

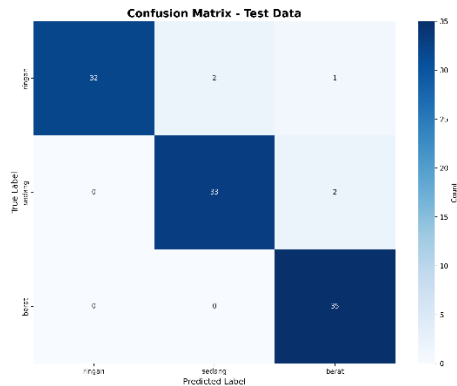
Tabel 15. Ringkasan Kinerja Model CNN

Dataset	Accuracy%	Loss
Data Latih	98.77%	0.0829
Data Validasi	93.43%	0.1953
Data Uji	95.24%	0.1612

Berdasarkan Tabel 15, model CNN menunjukkan performa yang sangat baik dengan nilai akurasi tertinggi pada data latih sebesar 98.71%, yang menunjukkan bahwa model mampu mempelajari pola fitur citra kerusakan jalan secara optimal. Pada data validasi, model memperoleh akurasi sebesar 93.43%, sedangkan pada data uji mencapai 95.24%, dengan nilai *loss* yang relatif rendah.

Selisih akurasi antara data latih, validasi, dan data uji tidak terlalu besar, yang menandakan bahwa model memiliki kemampuan generalisasi yang baik serta tidak mengalami *overfitting* yang signifikan.

Selain evaluasi kinerja secara keseluruhan, analisis hasil klasifikasi dilakukan menggunakan *confusion matrix* untuk mengetahui distribusi prediksi model pada setiap kelas kerusakan jalan. *Confusion matrix* model CNN terbaik ditunjukkan pada Gambar 15.



Gambar 17. Confussion Matrix

Berdasarkan *confusion matrix* dan laporan klasifikasi pada data uji, hasil evaluasi per kelas ditunjukkan pada Tabel 16.

Tabel 16. Hasil Evaluasi Klasifikasi Model CNN pada Data Uji

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
Ringan	100%	91.43%	95.52%	35
Sedang	89.47%	97.14%	93.15%	35
Berat	97.14%	97.14%	97.14%	35
<i>Accuracy%</i>			95.24%	105
Macro Average	95.54%	95.24%	95.27%	105
Weighted Average	95.54%	95.24%	95.27%	105

Berdasarkan hasil evaluasi menggunakan *confusion matrix*, model CNN memperoleh akurasi sebesar 95,24%. Dari total 105 citra uji, sebanyak 100 citra berhasil diklasifikasikan dengan benar, sedangkan 5 citra lainnya masih mengalami kesalahan prediksi. Nilai akurasi yang tinggi ini menunjukkan bahwa sebagian besar hasil prediksi model sudah sesuai dengan label sebenarnya.

Untuk kelas kerusakan ringan, model mencapai *precision* sebesar 100%, *recall* sebesar 91,43%, dan *F1-score* sebesar 95,52%. Nilai *precision* yang sempurna menandakan bahwa seluruh citra yang diprediksi sebagai kelas ringan memang benar berasal dari kelas tersebut. Namun, nilai *recall* sedikit lebih rendah karena masih ada beberapa citra ringan yang salah terdeteksi sebagai kelas lain.

Pada kelas kerusakan sedang, model memperoleh *precision* sebesar 89,47%, *recall* sebesar 97,14%, dan *F1-score* sebesar 93,15%. Nilai *recall* yang lebih tinggi menunjukkan bahwa sebagian besar citra kelas sedang berhasil dikenali dengan baik, meskipun masih ada sedikit kesalahan prediksi dari kelas lain.

Sedangkan untuk kelas kerusakan berat, model menghasilkan *precision* sebesar 97,14%, *recall* sebesar 97,14%, dan *F1-score* sebesar 97,14%. Hasil ini menunjukkan bahwa model mampu mengenali dan memprediksi citra kelas berat dengan tingkat ketepatan yang tinggi dan stabil.

Secara keseluruhan, model memiliki nilai *macro average precision* sebesar 95,54%, *recall* sebesar 95,24%, dan *F1-score* sebesar 95,27%. Nilai rata-rata yang seimbang ini menunjukkan bahwa performa model cukup baik di semua kelas, tanpa ada kelas tertentu yang terlalu mendominasi. Dengan demikian, model CNN dapat dikatakan mampu mengklasifikasikan tingkat kerusakan jalan dengan cukup akurat dan konsisten.

4.5.5. Hasil Pengujian Dengan Data Tambahan

Pada tahap ini, setiap citra pada data tambahan diproses menggunakan model CNN dengan parameter terbaik yang telah ditentukan sebelumnya. Model kemudian menghasilkan prediksi kelas tingkat kerusakan jalan aspal, yaitu ringan, sedang, atau berat, beserta nilai *confidence score* untuk setiap prediksi. Hasil prediksi tersebut dibandingkan dengan label sebenarnya untuk mengetahui apakah klasifikasi yang dilakukan oleh model sudah sesuai.

Hasil pengujian generalisasi model CNN menggunakan data tambahan disajikan dalam bentuk tabel yang memuat informasi nama file citra, kelas sebenarnya, hasil prediksi model, serta nilai *confidence score*. Hasil ini selanjutnya dianalisis untuk melihat konsistensi performa model.

Tabel 17 Hasil Pengujian Model CNN Menggunakan Data Tambahan

No	Gambar	Kelas Sebenarnya	Prediksi Model	<i>Confidence Score%</i>
1		Sedang	Sedang	95.6%
2		Berat	Berat	96.53%
3		Ringan	Ringan	95.62%

4		Sedang	Sedang	76.2%
5		Ringan	Ringan	91.2%
6		Berat	Berat	95.1%

Berdasarkan hasil pengujian tersebut, dapat dilihat bahwa sebagian besar citra pada data tambahan berhasil diklasifikasikan dengan benar oleh model CNN. Prediksi yang dihasilkan juga memiliki nilai *confidence score* yang relatif tinggi, terutama pada citra dengan tingkat kerusakan yang terlihat jelas. Hal ini

menunjukkan bahwa model CNN memiliki kemampuan generalisasi yang cukup baik terhadap data baru.

Namun demikian, pada beberapa citra tertentu masih ditemukan kesalahan klasifikasi, yang umumnya terjadi pada citra dengan karakteristik kerusakan yang mirip antar kelas, seperti perbedaan antara tingkat kerusakan ringan dan sedang. Kondisi ini menunjukkan bahwa performa model masih dipengaruhi oleh variasi visual dan kualitas citra *input*.

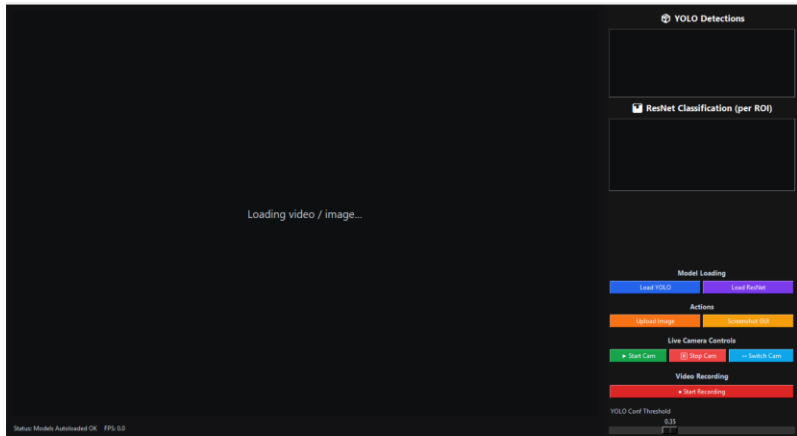
Secara keseluruhan, hasil pengujian dengan data tambahan menunjukkan bahwa model CNN tidak hanya mampu bekerja dengan baik pada dataset utama, tetapi juga tetap memberikan prediksi yang konsisten ketika diuji menggunakan citra baru. Hal ini mendukung penggunaan model CNN sebagai pengklasifikasi tingkat kerusakan jalan aspal pada kondisi nyata.

4.6. Implementasi Model Pada GUI

Pada tahap ini, model YOLO dan CNN yang telah dilatih sebelumnya diimplementasikan ke dalam sebuah aplikasi berbasis *Graphical User Interface* (GUI). Pembuatan GUI bertujuan untuk mempermudah pengguna dalam melakukan proses deteksi dan klasifikasi kerusakan jalan tanpa harus menjalankan kode secara manual. Dengan adanya antarmuka ini, proses analisis citra dapat dilakukan hanya dengan mengupload gambar atau menangkap citra secara langsung melalui webcam, sehingga aplikasi lebih mudah digunakan oleh pengguna awam maupun pihak yang membutuhkan proses inspeksi lapangan.

Dalam penelitian ini, pembuatan GUI menggunakan *framework* Tkinter, yaitu *library* bawaan Python yang sering digunakan untuk mengembangkan antarmuka aplikasi *desktop*. Pemilihan Tkinter didasarkan pada beberapa pertimbangan, antara lain kemudahan integrasi dengan model deep learning, ukuran aplikasi yang ringan, serta kemudahan dalam pengembangan dan kustomisasi tampilan.

Proses implementasi GUI dilakukan dengan mengintegrasikan dua model utama, yaitu model YOLO untuk mendeteksi area retakan, serta model CNN untuk melakukan klasifikasi tingkat kerusakan (ringan, sedang, atau berat). Kedua model ini dihubungkan dalam satu alur kerja sehingga hasil deteksi dapat langsung diproses ke tahap klasifikasi.



Gambar 18. Tampilan GUI

4.6.1. Hasil Pengujian Fungsional GUI

Tabel 18. Hasil Pengujian Fungsional

Jenis Pengujian	Skenario Uji	Output yang Diharapkan	Status
Tombol <i>Load</i> Model YOLO	Pengguna menekan tombol <i>Load</i> YOLO	Model YOLO berhasil dimuat dan status model siap digunakan	✓
Tombol <i>Load</i> Model Resnet	Pengguna menekan tombol <i>Load</i> ResNet	Model CNN (ResNet50) berhasil dimuat untuk klasifikasi ROI	✓
<i>Upload</i> Gambar	Pengguna menekan tombol <i>Upload Image</i> dan memilih file citra	Gambar tampil pada area utama GUI dan siap diproses	✓
Deteksi YOLO (statis)	Gambar statis diproses oleh model YOLO	Bounding box retak buaya muncul pada citra	✓
Klasifikasi CNN (statis)	ROI hasil deteksi YOLO dikirim ke CNN	Label kelas (Ringan/Sedang/Berat)	✓

		muncul di atas bounding box	
<i>Start Camera</i>	Pengguna menekan tombol <i>Start Cam</i>	Webcam aktif dan menampilkan video <i>realtime</i> pada GUI	✓
Deteksi YOLO <i>Realtime</i>	Kamera diarahkan ke permukaan jalan retak	Bounding box muncul secara <i>realtime</i> mengikuti objek retakan	✓
Klasifikasi CNN <i>Realtime</i>	Setiap ROI hasil deteksi <i>realtime</i> diklasifikasikan	Label kelas kerusakan tampil konsisten di atas <i>bounding box</i>	✓
<i>Stop Camera</i>	Pengguna menekan tombol <i>Stop Camera</i>	Webcam berhenti dan tampilan video dihentikan	✓
<i>Screenshot GUI</i>	Pengguna menekan tombol <i>Screenshot GUI</i>	Tampilan GUI tersimpan sebagai file gambar	✓
<i>Start Recording</i>	Pengguna menekan tombol <i>Start Recording</i>	Proses perekaman video <i>realtime</i> dimulai	✓
<i>Stop Recording</i>	Pengguna menekan tombol <i>Stop Recording</i>	Video hasil perekaman tersimpan di folder recording	✓
<i>Slider Confidence YOLO</i>	Pengguna mengubah nilai <i>confidence threshold</i>	Sensitivitas deteksi YOLO berubah sesuai nilai <i>threshold</i>	✓

Berdasarkan hasil pengujian pada Tabel 18, seluruh fitur utama pada aplikasi GUI dapat berfungsi dengan baik. Tombol pemuatan model YOLO dan CNN berhasil memuat model tanpa kendala, serta proses deteksi dan klasifikasi dapat berjalan baik pada mode citra statis maupun *realtime*. Selain itu, fitur pendukung seperti *screenshot GUI*, perekaman video, dan pengaturan *confidence threshold*

juga dapat digunakan sesuai dengan tujuan perancangan sistem. Hasil ini menunjukkan bahwa aplikasi GUI yang dikembangkan telah memenuhi kebutuhan fungsional dan siap digunakan sebagai media implementasi sistem deteksi dan klasifikasi kerusakan jalan.

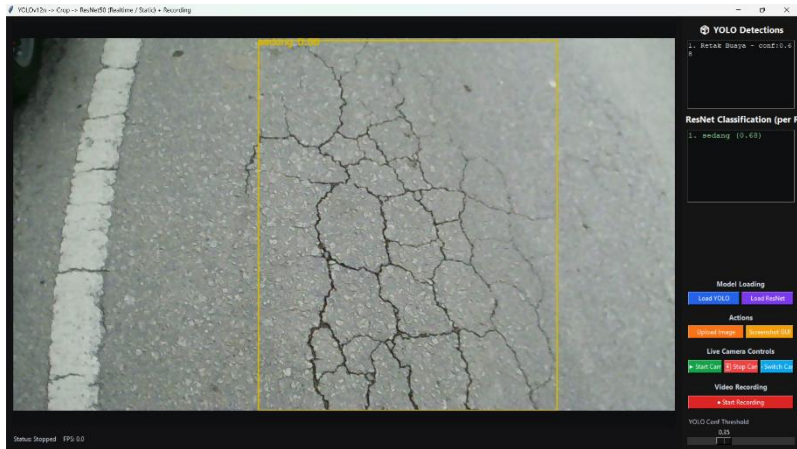
4.6.1 Hasil Pengujian Secara *Realtime*

Pengujian secara *realtime* dilakukan untuk memastikan bahwa integrasi antara model YOLO sebagai pendeteksi retak buaya dan model CNN sebagai pengklasifikasi tingkat kerusakan dapat berjalan dengan baik ketika digunakan langsung melalui *input* webcam. Pengujian ini juga bertujuan melihat stabilitas sistem saat digunakan dalam kondisi yang lebih mendekati penggunaan lapangan, yaitu ketika objek ditangkap secara langsung dan model harus melakukan proses deteksi dan klasifikasi secara otomatis.

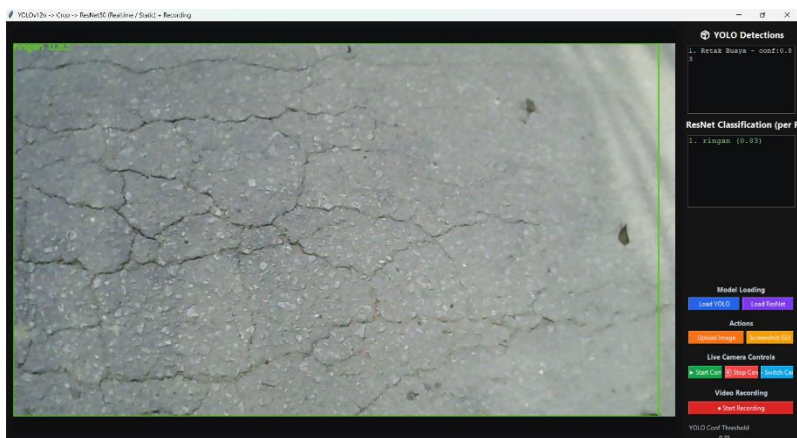
Pada pengujian ini, kondisi yang digunakan adalah pencahayaan terang (siang hari), karena kondisi tersebut paling umum dan paling optimal untuk kamera. Pengguna mengarahkan webcam ke permukaan jalan yang memiliki pola retak buaya, dan GUI akan menampilkan *bounding box* hasil deteksi YOLO sekaligus label kelas kerusakan dari CNN secara *realtime*.



Gambar 19. Hasil Deteksi dan Klasifikasi Realtime Kelas Berat



Gambar 20. Hasil Deteksi dan Klasifikasi Realtime Kelas Sedang



Gambar 21. Hasil Deteksi dan Klasifikasi Realtime Kelas Ringan

Untuk merangkum hasil pengujian, tabel berikut digunakan sebagai dasar evaluasi:

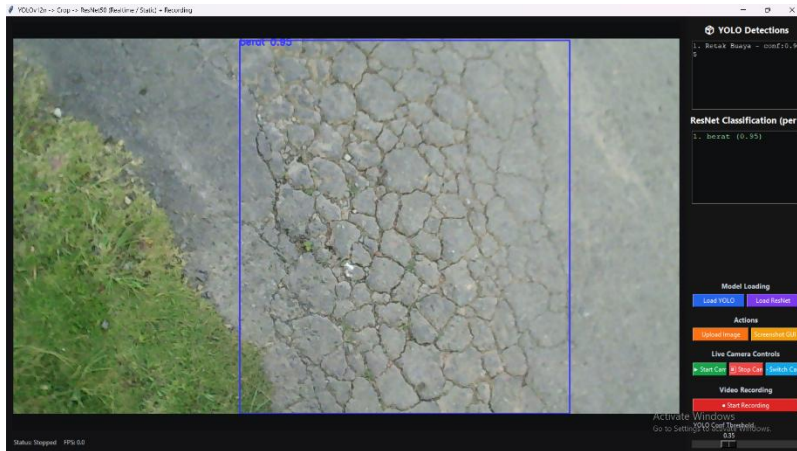
Tabel 19. Hasil Pengujian Gui Secara Realtime

No	Parameter yang Diamati	Hasil Pengamatan
	Sumber input	Webcam memberikan input stabil tanpa frame drop yang mengganggu proses deteksi.
	Objek retak buaya	Retakan berhasil terdeteksi dari beberapa jarak dan sudut kamera
	Deteksi YOLO	Bounding box tampil secara realtime, responsif, dan mengikuti posisi retakan dengan baik.
	Klasifikasi CNN	ROI hasil deteksi langsung diklasifikasikan menjadi Ringan/Sedang/Berat; label muncul tepat di atas <i>bounding box</i> .
	Output GUI	<i>Bounding box</i> , label kelas, dan FPS tampil dengan lancar tanpa lag signifikan.
	Perbandingan dengan gambar statis	Hasil <i>realtime</i> konsisten dengan prediksi citra statis, baik dari sisi deteksi maupun klasifikasi.

Berdasarkan hasil pengujian pada tabel 17 di atas, YOLO dapat menyesuaikan *bounding box* dengan cepat meskipun terdapat perubahan sudut atau pergerakan objek, menandakan bahwa model memiliki responsivitas yang baik. CNN juga memberikan hasil klasifikasi yang konsisten pada setiap ROI, menunjukkan bahwa proses *cropping* otomatis dari YOLO berhasil memfasilitasi *input* yang sesuai untuk klasifikasi.

Respon GUI yang lancar menunjukkan bahwa kombinasi model dan konfigurasi pemrosesan sudah optimal untuk penggunaan *realtime*. Hal ini menandakan bahwa sistem memiliki potensi untuk diterapkan dalam pengawasan kondisi jalan secara langsung, selama kondisi pencahayaan dan pergerakan kamera masih berada dalam batas yang wajar.

Dilakukan pengujian sistem deteksi dan klasifikasi retak buaya pada kondisi pencahayaan sore hari. Pengujian ini bertujuan untuk mengevaluasi kinerja model pada kondisi intensitas cahaya yang mulai menurun serta munculnya bayangan yang lebih dominan akibat sudut datang cahaya matahari yang rendah.

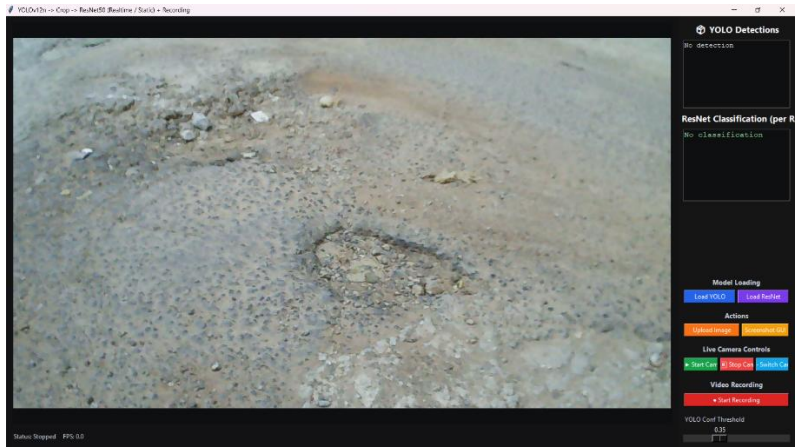


Gambar 22. Hasil Pengujian Pada Kondisi Pencahayaan Sore Hari

Berdasarkan hasil pengujian yang ditunjukkan pada Gambar 22, model YOLO mampu mendeteksi retak buaya dengan baik yang dilakukan pada sore hari. Area retakan ditandai dengan *bounding box* secara tepat, dan hasil deteksi tersebut dapat diklasifikasikan oleh model CNN.

Keberhasilan model dalam mendeteksi dan mengklasifikasikan retak buaya pada kondisi pencahayaan sore hari menunjukkan bahwa perubahan intensitas cahaya tidak secara langsung menurunkan kemampuan sistem dalam mengenali pola retakan. Hal ini mengindikasikan bahwa kondisi pencahayaan sore hari tidak memberikan pengaruh signifikan terhadap performa sistem. Kemampuan tersebut didukung oleh penerapan *preprocessing* dan augmentasi data pada tahap pelatihan, sehingga model memiliki kemampuan generalisasi yang baik terhadap variasi pencahayaan. Dengan demikian, sistem deteksi dan klasifikasi retak buaya yang dikembangkan tetap dapat bekerja secara andal pada kondisi pencahayaan sore hari meskipun intensitas cahaya lebih rendah dibandingkan kondisi siang hari.

Sebagai pengujian tambahan, dilakukan pula percobaan terhadap citra yang menampilkan jenis kerusakan jalan lain, yaitu lubang (*pothole*). Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem hanya mendeteksi dan mengklasifikasikan objek sesuai dengan kelas yang telah dilatih, yaitu retak buaya.

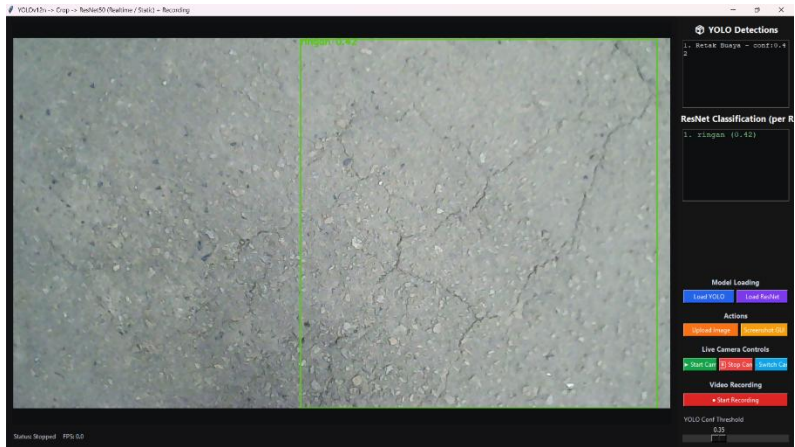


Gambar 23. Hasil Pengujian Pada Jalan Lubang

Berdasarkan pada Gambar 23 menunjukkan bahwa model tidak memberikan output deteksi maupun klasifikasi pada citra atau objek jalan berlubang. Hal ini menandakan bahwa sistem bekerja sesuai dengan tujuan awal, yaitu hanya mendeteksi pola retak buaya dan mengabaikan jenis kerusakan lain yang tidak termasuk dalam dataset pelatihan. Dengan demikian, model memiliki spesifisitas yang baik terhadap kelas yang dilatih dan tidak melakukan prediksi pada objek yang tidak relevan.

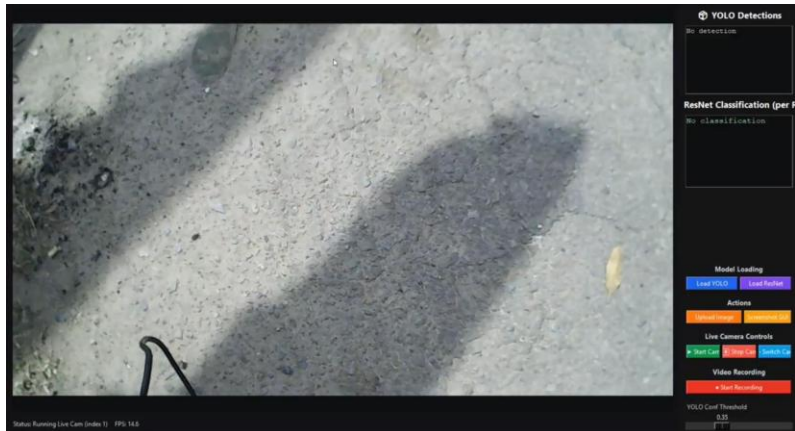
Selain pengujian pada citra dengan ukuran retakan yang jelas dan dominan, dilakukan pula pengujian terhadap citra yang memiliki pola retakan berukuran kecil. Berdasarkan hasil pengujian, diketahui bahwa model tidak selalu mampu mendeteksi retakan berukuran kecil secara konsisten.

Pada beberapa citra, retakan kecil berhasil terdeteksi dengan baik, di mana *bounding box* muncul pada area retakan sebagaimana ditunjukkan pada Gambar 23.



Gambar 24. Hasil Deteksi Pada Retakan Berukuran Kecil

Namun, pada citra dengan pola retakan berukuran kecil, model tidak selalu konsisten dalam melakukan deteksi. Pada beberapa kali percobaan, retakan berhasil terdeteksi, sedangkan pada percobaan lain tidak muncul hasil deteksi, seperti yang diperlihatkan pada Gambar 25. Kondisi ini terjadi karena nilai *confidence score* yang dihasilkan terlalu rendah, sehingga model YOLO kurang yakin bahwa area tersebut merupakan retak buaya. Akibatnya, model CNN juga tidak dapat melakukan klasifikasi karena tidak menerima *input* hasil deteksi dari YOLO.



Gambar 25. Hasil Deteksi Pada Retakan Berukuran Kecil

Retakan dengan ukuran kecil dan pola tipis ini umumnya termasuk ke dalam kategori retak ringan, di mana tingkat keparahan kerusakan masih rendah dan permukaan jalan belum menunjukkan deformasi signifikan. Dengan demikian, dapat disimpulkan bahwa ukuran fisik retakan pada citra, khususnya pada kelas retak ringan, berpengaruh terhadap keberhasilan deteksi. Model cenderung lebih akurat mendeteksi retak buaya dengan ukuran yang lebih besar dan pola yang tegas, seperti pada kelas sedang dan berat.

Bab 5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan seluruh proses penelitian yang telah dilakukan, dapat disimpulkan bahwa sistem deteksi dan klasifikasi kerusakan pada jalan aspal yang dibangun dalam penelitian ini mampu bekerja dengan sangat baik. Model YOLOv12n yang digunakan sebagai pendeteksi retakan menunjukkan performa yang stabil dengan nilai mAP50 sebesar 90,08%. Nilai ini menunjukkan bahwa model cukup handal dalam mengenali pola-pola retakan pada berbagai jenis citra. Hal tersebut juga terlihat pada pengujian *realtime*, di mana *bounding box* dapat muncul secara tepat dan tetap mengikuti posisi retakan meskipun sudut kamera berubah.

Untuk tahap klasifikasi tingkat kerusakan, model CNN dengan arsitektur ResNet50 juga memberikan hasil yang sangat memuaskan. Berdasarkan hasil evaluasi menggunakan confusion matrix, model mencapai akurasi sebesar 95,24%, dengan nilai *precision*, *recall*, dan *F1-score* yang merata pada ketiga kelas (Ringan, Sedang, dan Berat). Hasil ini membuktikan bahwa model mampu membedakan tingkat keparahan retak secara konsisten dan akurat.

Integrasi kedua model tersebut ke dalam GUI juga berjalan lancar. GUI dapat menampilkan hasil deteksi YOLO dan klasifikasi CNN secara bersamaan, baik ketika menggunakan gambar maupun saat berjalan *realtime* lewat webcam. Selama pengujian, aplikasi dapat berjalan tanpa gangguan berarti, dengan tampilan *bounding box*, label kelas, dan FPS yang muncul secara stabil. Secara keseluruhan, penelitian ini menunjukkan bahwa metode YOLO dapat digunakan sebagai alat bantu deteksi kerusakan jalan dengan cepat dan akurat, sementara model CNN mampu mengklasifikasikan tingkat keparahan retak dengan performa yang sangat baik. Integrasi kedua model ini ke dalam aplikasi GUI menghasilkan sistem yang mudah digunakan dan dapat melakukan deteksi serta klasifikasi secara *realtime*.

5.2. Saran

Melalui penelitian ini, dapat disimpulkan bahwa sistem deteksi dan klasifikasi kerusakan jalan yang dikembangkan sudah menunjukkan hasil yang cukup baik, meskipun masih ada beberapa hal yang bisa diperbaiki. Beberapa faktor seperti proses pengambilan data dan kondisi lingkungan saat perekaman masih bisa ditingkatkan supaya model bisa bekerja lebih optimal di berbagai kondisi nyata.

Kedepannya, pengembangan dataset menjadi hal penting yang perlu dilakukan. Menambahkan variasi kondisi pencahayaan, seperti citra pada sore hari, malam hari, atau saat cuaca mendung, dapat membantu model menjadi lebih tangguh (*robust*). Selain itu, memperluas jenis kerusakan jalan misalnya dengan

menambahkan kelas seperti lubang (*potholes*), retak memanjang, retak melintang, atau kerusakan permukaan lainnya yang dapat membuat sistem ini lebih lengkap.

Sistem ini diharapkan bisa terus dikembangkan agar nantinya dapat digunakan sebagai alat bantu dalam kegiatan perawatan dan pemeliharaan jalan oleh pihak-pihak seperti Binamarga atau Dinas Pekerjaan Umum (DPU). Dengan pengembangan lebih lanjut, sistem ini berpotensi menjadi solusi yang efektif untuk mempercepat proses pemeriksaan kondisi jalan secara otomatis dan efisien.

Daftar Pustaka

- [1] GoodStats, "Kemantapan jalan nasional capai 94,2% pada 2023," <https://data.goodstats.id/statistic/kemantapan-jalan-nasional-capai-942-pada-2023-102eb>. Accessed: Jul. 13, 2025. [Online]. Available: <https://data.goodstats.id/statistic/kemantapan-jalan-nasional-capai-942-pada-2023-102eb>
- [2] I. Salamah, M. R. A. Said, and S. Soim, "Perancangan Alat Identifikasi Wajah Dengan Algoritma You Only Look Once (YOLO) Untuk Presensi Mahasiswa," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 3, p. 1492, Jul. 2022, doi: 10.30865/mib.v6i3.4399.
- [3] L. G. Denaro and R. Lim, "Analisis Deteksi Kerusakan pada Jalan Aspal menggunakan Deep Learning untuk Mendukung Efisiensi Biaya dan Waktu dalam Pemantauan Berkelanjutan," *Jurnal Dimensi Insinyur Profesional*, vol. 3, no. 1, pp. 16–25, Mar. 2025, doi: 10.9744/jdip.3.1.16-25.
- [4] B. Sasmito, B. H. Setiadji, and R. Isnanto, "Deteksi Kerusakan Jalan Menggunakan Pengolahan Citra Deep Learning di Kota Semarang," *TEKNIK*, vol. 44, no. 1, pp. 7–14, May 2023, doi: 10.14710/teknik.v44i1.51908.
- [5] H. Yunardhi, M. J. Alkasi, and H. Sutanto, "JURNAL TEKNOLOGI SIPIL."
- [6] R. Rahmawati and A. Maruman, "STUDI ANALISIS KINERJA JALAN RAYA AKIBAT BEBAN LALU LINTAS DAN FAKTOR LINGKUNGAN."
- [7] L. G. Denaro and R. Lim, "Analisis Deteksi Kerusakan pada Jalan Aspal menggunakan Deep Learning untuk Mendukung Efisiensi Biaya dan Waktu dalam Pemantauan Berkelanjutan," *Jurnal Dimensi Insinyur Profesional*, vol. 3, pp. 16–25, Mar. 2025, doi: 10.9744/jdip.3.1.16-25.
- [8] Y. Sari1, A. Rizky Baskara, B. Prakoso, and M. A. Rahman, "PENERAPAN ACTIVE CONTOUR MODEL PADA PENGOLAHAN CITRA UNTUK DETEKSI KERUSAKAN JALAN (APPLICATION OF ACTIVE CONTOUR MODEL ON IMAGE PROCESSING FOR DETECTION OF ROAD DAMAGE)."
- [9] M. Fikri, "Analisis Tingkat Kerusakan Jalan Lentur dengan Metode Pavement Condition Index (PCI)... ANALISIS TINGKAT KERUSAKAN JALAN LENTUR DENGAN METODE PAVEMENT CONDITION INDEX (PCI) STUDI KASUS RUAS JALAN POROS LAMASI-WALENRANG KABUPATEN LUWU."
- [10] J. Nurhakiki *et al.*, "Studi Kepustakaan: Pengenalan 4 Algoritma Pada Pembelajaran Deep Learning Beserta Implikasinya," *Jurnal Pendidikan Berkarakter*, no. 1, pp. 270–281, 2024, doi: 10.51903/pendekar.v2i1.598.
- [11] A. Raup, W. Ridwan, Y. Khoeriyah, Q. Yuliati Zaqiah, and U. Islam Negeri Sunan Gunung Djati Bandung, "Deep Learning dan Penerapannya dalam Pembelajaran." [Online]. Available: <http://jiip.stkipyapisdompau.ac.id>

- [12] T. Perumal, N. Mustapha, R. Mohamed, and F. M. Shiri, "A Comprehensive Overview and Comparative Analysis on Deep Learning Models," *Journal on Artificial Intelligence*, vol. 6, no. 1, pp. 301–360, 2024, doi: 10.32604/jai.2024.054314.
- [13] K. N. Nair *et al.*, "Modified YOLOv4 for real-time Coconut Trees Detection from an Unmanned Aerial Vehicle," Aug. 20, 2021. doi: 10.21203/rs.3.rs-826223/v1.
- [14] E. Royhan, M. Beta, A. Sucipto, R. Regasari, M. Putri, and B. D. Setiawan, "PENGEMBANGAN SISTEM DETEKSI LUBANG PADA JALAN MENGGUNAKAN ALGORITMA YOLO BERBASIS ESP32-CAM," 2025. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [15] D. Raya Ismail, R. Rahmadewi Teknik Elektro, U. Singaperbangsa Karawang HSRonggo Waluyo, K. Telukjambe Tim, K. Karawang, and J. Barat, "SISTEM DETEKSI JALAN BERLUBANG SECARA REAL-TIME MENGGUNAKAN YOLOV11: INTEGRASI DATA DAN LOKASI MELALUI WEBSITE (STUDI KASUS : DAERAH KARAWANG)," 2025.
- [16] A. Fajar, D. Iskandar Mulyana, A. Septian Amrullah, K. Irfan Nauval, and T. Informasi Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, "Identifikasi Kerusakan Jalan dengan Metode Faster R-CNN Studi Kasus di Jalan Pakansari Bogor Jawa barat," vol. 11, no. 2, 2022.
- [17] M. I. Mardiyah, "Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) Pada Citra Kebun Dan Sawah (Studi Kasus : Klasifikasi Gambar Pada Citra Sawah dan Kebun)," <https://dspace.uui.ac.id/handle/123456789/28083>.
- [18] "Convolutional Neural Network (CNN) dalam Pengolahan Citra Digital," <https://psti.unisayogya.ac.id/2025/05/19/convolutional-neural-network-cnn-dalam-pengolahan-citra-digital/>.
- [19] - DODI EFENDI, "PENERAPAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK ARSITEKTUR RESNET-50 UNTUK KLASIFIKASI CITRA DAGING SAPI DAN BABI," 2022.
- [20] "View of ResNet50 architecture performance enhancements to Address Overfitting Issues in Skin Disease Classification." Accessed: Jun. 17, 2025. [Online]. Available: <https://jurnal.plb.ac.id/index.php/tematik/article/view/1876/885>
- [21] "Image Segmentation and Classification in Skin Cancer Using Convolutional Neural Network (CNN) Method with ResNet-50 Architecture | Repostor's Journal." Accessed: Jun. 17, 2025. [Online]. Available: <https://ejournal.umm.ac.id/index.php/repositor/article/view/34415?articlesBySameAuthorPage=2>
- [22] R. R. P. Yanni, I. Mardiyah, D. C. Irawati, R. Kosasih, and D. P. Sari, "PERBANDINGAN KLASIFIKASI KERUSAKAN JALAN MODEL CNN VGG19

- DAN RESNET50," *Jurnal Ilmiah Informatika Komputer*, vol. 30, no. 1, pp. 69–80, 2025, doi: 10.35760/ik.2025.v30i1.14230.
- [23] A. R. Hermanto, A. Aziz, and S. Sudianto, "Perbandingan Arsitektur MobileNetV2 dan RestNet50 untuk Klasifikasi Jenis Buah Kurma," vol. 12, no. 4, 2024, doi: 10.26418/justin.v12i4.80358.
 - [24] 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.
 - [25] K. Azmi, S. Defit, and U. Putra Indonesia YPTK Padang Jl Raya Lubuk Begalung-Padang-Sumatera Barat, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," vol. 16, no. 1, p. 2023.
 - [26] 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.
 - [27] M. Mega Nugraha, M. Dani, F. Saputra, D. A. Fauzan, and E. Y. Puspaningrum, "Augmentasi Data Pengenalan Citra Batik Yogyakarta Menggunakan Pendekatan Random Crop, Rotate, Dan Mixup," *Prosiding Seminar Nasional Informatika Bela Negara (SANTIKA)*, vol. 5, no. 1, pp. 152–158, Aug. 2025, doi: 10.33005/SANTIKA.V5I1.680.
 - [28] B. Nugroho and E. Yulia, "KINERJA METODE CNN UNTUK KLASIFIKASI PNEUMONIA DENGAN VARIASI UKURAN CITRA INPUT," vol. 8, no. 3, pp. 533–538, 2021, doi: 10.25126/jtiik.202184515.

Biodata



Nama : Farhan Muhammad Yusuf
TTL : Batam, 29-09-2004
Agama : Islam
Alamat : KAVLING SUMBER SERAYA B14/29

Email : farhanmyusuf362@gmail.com
Riwayat Pendidikan SMA : SMA NEGERI 17 BATAM
SMP : SMP NEGERI 21 BATAM

Lampiran

https://drive.google.com/drive/folders/1zolzwa5hWgaZ6dlxWEKc_uPKyMy00Gf?usp=sharing