

SKRIPSI

PEMANFAATAN TEKNOLOGI *DEEP LEARNING* DALAM DETEKSI TINGKAT KERUSAKAN JALAN MENGGUNAKAN CNN (Studi Kasus: kecamatan Wasile, Kabupaten Haltim, Maluku Utara)



OLEH
Rinaldi Abdul
07352011069

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KHAIRUN
TERNATE
2024**

LEMBAR PENGESAHAN

PEMANFAATAN TEKNOLOGI *DEEP LEARNING* DALAM DETEKSI TINGKAT KERUSAKAN JALAN MENGGUNAKAN CNN (Studi Kasus: Kecamatan Wasile, Kabupaten Haltim, Maluku Utara)

Oleh
Rinaldi Abdul
07352011069

Skripsi ini telah disahkan
Tanggal 24 Juli 2024

Menyetujui
Tim Penguji

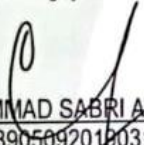
Ketua Penguji


ROSIHAN, S.T., M.Cs.
NIP. 197607192010121001

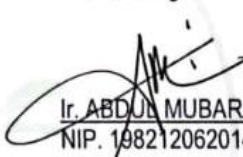
Pembimbing I


Ir. AMAL KHAIRAN, S.T., M.Eng., IPM.
NIP. 197401112003121003

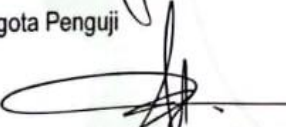
Anggota Penguji


MUHAMMAD SABRI AHMAD, S.Kom., M.Kom.
NIP. 198905092013031013

Pembimbing II


Ir. ABDUL MUBARAK, S.Kom., M.T., IPM.
NIP. 198212062014041002

Anggota Penguji


SYARIFUDDIN M. KAPITA, S.Pd., M.Si.
NIP. 199103122024211001

Mengetahui/Menyetujui

Koordinator Program Studi
Informatika


ROSIHAN, S.T., M.Cs.
NIP. 197607192010121001


Dekan Fakultas Teknik
Universitas Khairun
Ir. ENDAH HARISUN, S.T., M.T., CRP.
NIP. 197511302015011013

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Rinaldi Abdul
NPM : 07352011069
Fakultas : Teknik
Jurusan/Program Studi : Informatika
Judul Skripsi : Pemanfaatan Teknologi *Deep Learning* Dalam Deteksi
Tingkat Kerusakan Jalan Menggunakan CNN (Studi Kasus:
Kecamatan Wasile, Kabupaten Haltim, Maluku Utara)

Dengan ini menyatakan bahwa penulisan Skripsi yang telah saya buat ini merupakan hasil karya sendiri dan benar keasliannya. Apabila ternyata di kemudian hari penulisan Skripsi ini merupakan hasil plagiat terhadap karya orang lain, maka saya bersedia mempertanggung jawabkan sekaligus bersedia menerima sanksi berdasarkan aturan tata tertib di Universitas Khairun.

Demikian pernyataan ini saya buat dalam keadaan sadar dan tidak dipaksakan.

Penulis



Rinaldi Abdul

HALAMAN PERSEMBAHAN

Tiada lembar skripsi yang paling indah dalam laporan ini kecuali lembar persembahan, *bismillahirrahmannirrahim* atas segala Rahmat Allah *Subhanahu Wa Ta'ala* yang Maha Pengasih lagi Maha Penyayang serta mengucapkan rasa syukur *Alhamdulillah* atas nikmat yang di berikan tanpa hentinya, penulis persembahkan skripsi ini kepada:

1. Bapak Kamaruddin Abdul dan Sarni Nuhi, kedua orang tua hebat yang telah senantiasa memberikan kasih sayang, nasihat, motivasi dan dukungan, serta doa kepada penulis sehingga dapat menyelesaikan penulisan skripsi ini. Semoga beliau selalu bangga kepada penulis. Terima kasih mama dan papa.
2. Terimakasih kepada saudara dan saudari saya yang bernama Rachmawati, Susmiati, Muhammad Riski, yang telah memberikan dukungan kepada saya, sehingga dapat menyelesaikan tugas akhir saya ini (Skripsi)
3. Terima kasih kepada teman-teman komplek yang selalu memberikan dukungan dan dorongan dalam perjalanan ini.
4. Terima kasih untuk teman-teman Panda Squad yaitu Suci Ayu Maharani, Sasmita Hi. Sadek, Lisa Elisia Potale, Nurwana Iswan, Aprilia Silawane, Harlina Sapsuha, Nafra Aziqra Hi. A, Marhama Maynaka, Cindy Rahawaty S. Hipy, Muhammad Raihan Rizal, Nirwandi Bahri, wahyudin, dan Ferliy, karena telah membantu dan menyemangati saya dalam menyelesaikan skripsi ini.
5. Terima kasih kepada dosen dosen yang telah membantu dan mengarahkan saya dalam penulisan skripsi ini sehingga dapat menyelesaikan skripsi ini.
6. Terima kasih kepada kak Satria Dwi Surya S.T., M. Kom., sebagai kakak yang selalu membantu dan mengarahkan saya selama ini.
7. Terakhir, Terima kasih kepada diri sendiri yang selalu semangat mampu berusaha keras dan berjuang sejauh ini untuk menyelesaikan skripsi ini, Terima kasih telah mengandalkan diri sendiri untuk tetap kuat tanpa bergantung kepada orang lain dan tidak pernah memutuskan untuk menyerah dalam situasi apapun.

MOTTO

“Terus Mengeluh Hanya Akan menunjukan seberapa lemahnya Dirimu”

KATA PENGANTAR

Puji syukur saya panjatkan kepada Allah *Subhanahu Wata'ala* yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga penulis dapat menyusun skripsi ini dengan judul "Pemanfaatan teknologi *Deep Learning* Dalam Deteksi tingkat Kerusakan Jalan Menggunakan CNN(Studi Kasus: Kecamatan Wasile, Kabupaten Halmahera Utara), ini dapat diselesaikan untuk memenuhi salah satu persyaratan penyelesaian pendidikan sarjana Informatika Strata Satu (S1) pada Program Studi Informatika Fakultas Teknik Universitas Unkhair.

Untuk menyelesaikan skripsi ini penulis sepenuhnya mendapat dukungan dari banyak pihak, oleh karena itu dengan rendah hati penulis mengucapkan terimakasih kepada:

1. Bapak Dr. Ridha Ajam, M.Hum., selaku Rektor Universitas Khairun Ternate.
2. Bapak Ir. Endah Harisun, S.T., M.T., CRP., selaku dekan Fakultas Teknik Universitas Khairun.
3. Bapak Rosihan, S.T., M.Cs., selaku Koordinator Program Studi Informatika Fakultas Teknik Universitas Khairun.
4. Bapak Ir. Amal Khairan, S.T., M.Eng. IPM., sebagai Pembimbing I, terima kasih atas bimbingannya, serta dukungan dalam penyelesaian skripsi ini.
5. Bapak Ir. Abdul Mubarak, S.Kom., M.T., IPM., sebagai Pembimbing II, terima kasih atas bimbingannya, serta dukungan dalam penyelesaian skripsi ini.
6. Kepada Kedua orang tua dan kakak yang senantiasa memberikan dukungan dan kasih sayang yang senantiasa dilimpahkan kepada penulis hingga saat ini.
7. Serta kepada teman-teman saya yang telah membantu serta memberikan dukungan sampai pada saat ini.
8. Bapak Rosihan, S.T., M.Cs, selaku penguji I yang telah memberikan masukan, Kritikan dan saran demi perbaikan skripsi ini.
9. Bapak muhammad Sabri Ahmad, S.Kom., M.Kom. selaku penguji II yang telah Memberikan Masukan, Kritikan dan saran demi Skripsi ini.
10. Bapak Syarifuddin N. Kapita, S.Pd., M.Si. selaku penguji III yang telah Memberikan Masukan, Kritikan dan saran demi skripsi ini.

11. Kedua orang tua penulis yang telah memberikan segenap cinta dan kasih sayang, yang tak henti-hentinya memberikan doa terbaik dan menjadi *support system* yang sehat bagi penulis. Semoga Allah *Subhanahu wa Ta'ala* memberikan rahmat, hidayah serta keberkahan kepada kita semua.
12. Rekan-rekan mahasiswa terutama panda squad yaitu Suci Ayu Maharani, Sasmita Hi. Sadek, Lisa Elisia Potale, Nurwana Iswan, Aprilia Silawane, Harlina Sapsuha, Nafra Aziqra Hi. A, Marhama Maynaka, Cindy Rahawaty S. Hipy, Muhammad Raihan Rizal, Nirwandi Bahri, dan Ferliy yang telah berpartisipasi baik saran, masukan, maupun kritik atas skripsi ini

Walaupun demikian dalam skripsi ini, penulis menyadari masih belum sempurna. oleh karena itu harapan penulis dalam memberikan kritik dan saran dari semua pihak yang bersifat membangun.

Ternate, 24 Juli 2024

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN KEASLIAN	iii
HALAMAN PERSEMBAHAN.....	iv
KATA PENGANTAR.....	v
DAFTAR ISI	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xi
ABSTRAK	xii
 BAB I PENDAHULUAN	
1.1. Latar belakang.....	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	4
1.6. Sistematika Penulisan	4
 BAB II TINJAUAN PUSTAKA	
2.1. Penelitian Terkait.....	6
2.2. Pengertian Jalan	10
2.3. Ruas Jalan Subaim	11
2.4. Kerusakan Jalan.....	11
2.5. Jenis-jenis kerusakan Jalan.....	12
2.6. <i>Deep Learning</i>	14
2.7. <i>Neural Network</i>	16
2.8. <i>Convolution Neural Network (CNN)</i>	18
2.8.1. <i>Input Layer</i>	19
2.8.2. <i>Feature Learning</i>	21
2.8.3. <i>Classification</i>	24

2.9. Fitmodel.....	26
2.10. Evaluasi Model	26
2.11. <i>Python</i>	28
2.12. Langkah Langkah Penggunaan <i>Python</i>	28
2.13. <i>Library Python</i>	30
2.14. <i>Flowchart</i>	31

BAB III METODE PENELITIAN

3.1. Objek dan Waktu Penelitian	33
3.2. Sifat Jenis dan Pendekatan Penelitian	33
3.3. Diagram Alir.....	34
3.4. Alat dan Bahan Penelitian	43
3.4.1. Perangkat keras (<i>Hardware</i>)	43
3.4.2. Perangkat Lunak (<i>Software</i>)	43

BAB IV HASIL DAN PEMBAHASAN

4.1. Analisis Data	45
4.2. <i>Preprosesing</i>	45
4.3. Implementasi CNN	48
4.4. Pelatihan Model.....	49
4.5. Evaluasi Model	51
4.6. <i>Deployment Model</i>	52

BAB V PENUTUP

5.1. Kesimpulan.....	57
5.2. Saran.....	57

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Ruas Jalan Subaim (<i>google Maps</i> , 2024)	11
Gambar 2.2. <i>Aligator Cracking</i> (Prametsya, 2018).	12
Gambar 2.3. <i>Corrugtion</i> (Prametsya, 2018).....	13
Gambar 2.4. <i>Despresion</i> (Prametsya, 2018)	13
Gambar 2.5. Lubang (Prametsya, 2018)	14
Gambar 2.6. Hubungan <i>Deep Learning</i> Dengan <i>Machine Learning</i> (Raup, 2022)	15
Gambar 2.7. Klasifikasi Dengan <i>Deep Learning</i> (Yoga Triardhana, 2020)	15
Gambar 2.8. Arsitektur <i>Neural Network</i> (Hadianto, 2019)	16
Gambar 2.9. Representasi <i>Neural Network</i> (Hadianto, 2019)	17
Gambar 2.10. Rumus dan Interpretasi <i>Neural Network</i> (Hadianto, 2019)	17
Gambar 2.11. Arsitektur Algoritma CNN (Nugroho, 2020).....	19
Gambar 2.12. <i>Input Layer</i> RGB (Sofyan, 2019).....	19
Gambar 2.13. Arsitektur tiga dimensi CNN (Zahra, 2023)	20
Gambar 2.14. Operasi <i>Convolution</i> citra menggunakan sebuah <i>filter</i> atau kernel (Saputra, 2022).....	22
Gambar 2.15. Pengaruh <i>Stirde</i> Terhadap operasi <i>Convolution</i> citra (Saputra, 2022).....	22
Gambar 2.16. Penambahan padding terhadap operasi <i>convolution</i> (Saputra, 2022).	23
Gambar 2.17. <i>Max Pooling</i> dan <i>Average Pooling</i> (Saputra, 2022)	24
Gambar 2.18. Fungsi Aktivasi <i>Rectified Linear Unit</i> (RELU) (Saputra, 2022).....	24
Gambar 2.19. <i>Fully Connected Layer</i> (Saputra, 2022)	25
Gambar 2.20. <i>Python</i>	28
Gambar 3.1. Diagram Alur Penelitian	34
Gambar 3.2. Proses Pengolahan Data	35
Gambar 3.3. <i>Convolution Matrix</i>	37
Gambar 3.4. <i>Convolution</i> Pada Posisi Ke1.....	38
Gambar 3.5. <i>Convolution</i> Pada Posisi Ke2.....	38
Gambar 3.6. Hasil Dari <i>Convolution</i>	38
Gambar 3.7. Hasil <i>Pooling Layer</i>	39

Gambar 3.8. Hasil Konvolusi menjadi vektor 1D.....	40
Gambar 4.1. <i>Dataset</i> Yang digunakan.....	45
Gambar 4.2. <i>Preprocessing Data</i>	47
Gambar 4.3. <i>preprocessing widthshift Range</i>	47
Gambar 4.4. Implementasi Model CNN	48
Gambar 4.5. Grafik Hasil Iterasi Nilai Akurasi	50
Gambar 4.6. Grafik Hasil Iterasi Nilai Los	51
Gambar 4.7. <i>confusion matrix</i>	52
Gambar 4.8. Implementasi <i>Library Python</i>	53
Gambar 4.9. Memuat Model CNN	53
Gambar 4.10. Kode sumber AntarMuka <i>Streamlit</i>	54
Gambar 4.11. Kode Sumber <i>File Uploader</i>	54
Gambar 4.11. Membuat <i>Input Button</i>	55
Gambar 4.12. Tampilan <i>Title, Text Input</i>	55
Gambar 4.13. Tampilan Memuat Gambar	56
Gambar 4.14. Tampilan Hasil Deteksi Kerusakan Jalan.....	56

DAFTAR TABEL

	Halaman
Tabel 2.1. Penelitian Terkait	5
Tabel 2.2. Confusion Matrix (hu, 2020).....	27
Tabel 2.2. Simbol dan Fungsi <i>Flowchart</i> (chairs Noiija,2023).....	31
Tabel 3.1. Detail Spesifikasi <i>Hardware</i>	43
Tabel 3.2. Spesifikasi <i>Software</i>	43

ABSTRAK

PEMANFAATAN TEKNOLOGI *DEEP LEARNING* DALAM DETEKSI TINGKAT KERUSAKAN JALAN MENGGUNAKAN CNN (Studi Kasus: kecamatan Wasile, Kabupaten Haltim, Maluku Utara)

Rinaldi Abdul¹, Amal Khairan², Abdul Mubarak³

Program Studi Informatika, Fakultas Teknik, Universitas Khairun

Jl. Jati Metro, Kota Ternate Selatan

Email: ¹RinaldiAbdul@gmail.com, ²4malkhairan@unkhair.ac.id, ³AbdulMubarak@gmail.com

Penelitian ini membahas deteksi tingkat kerusakan jalan menggunakan metode *Deep Learning* untuk menganalisis kondisi jalan dan memberikan alternatif jalur berdasarkan tingkat kerusakan. Studi kasus dilakukan di ruas jalan Subaim-Wasile, Kabupaten Haltim, yang merupakan penghubung antar-kabupaten di Provinsi Maluku Utara. Jalan di daerah ini mengalami berbagai jenis kerusakan seperti lubang, gelombang, dan retakan. Metode yang digunakan dalam penelitian ini adalah *Convolutional Neural Network* (CNN) yang diimplementasikan pada citra jalan yang diambil dengan kamera digital. Tujuan Penelitian ini untuk menciptakan model dengan tingkat akurasi tinggi dalam mendeteksi dan mengklasifikasikan kerusakan jalan, guna memudahkan pemeliharaan dan perbaikan jalan secara efektif dan efisien. Hasil penelitian menunjukkan bahwa model CNN yang diterapkan memiliki tingkat akurasi 88% dalam mendeteksi kerusakan jalan, yang diharapkan dapat membantu otoritas terkait dalam mempercepat proses identifikasi dan penanganan kerusakan jalan.

Kata kunci: *Deep Learning*, *Convolutional Neural Network*, deteksi kerusakan jalan, Subaim Wasile, pemeliharaan jalan.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Jalan merupakan infrastruktur yang penting, memiliki peran dalam mendukung berlangsungnya kelancaran aktivitas yang dilakukan oleh pengguna jalan. Kondisi infrastruktur jalan yang baik dapat memberikan rasa aman dan nyaman kepada pengguna jalan, di sisi lain jalan yang rusak tentunya dapat menghambat kegiatan pengguna jalan dan dapat menjadi penyebab terjadinya kecelakaan (Lalu, 2022).

Saat ini kerusakan jalan masih banyak terjadi pada jalan-jalan yang ada di Indonesia. Seringkali kita menemui kondisi beberapa ruas jalan dalam keadaan rusak seperti pada umumnya kerusakan lubang, kerusakan bergelombang, kerusakan retak dan beberapa jenis kerusakan lainnya. Kerusakan jalan yang ditemui pada jalan-jalan yang ada diberbagai daerah di Indonesia perlu mendapatkan perhatian khusus. Hal ini menjadi penting dikarenakan kerusakan jalan akan memberikan dampak secara langsung bagi pengguna jalan, baik dari segi ekonomi, waktu tempuh dan keselamatan (Rinaldi, 2022).

Kerusakan yang terjadi pada jalan dapat disebabkan oleh faktor – faktor yang merusak jalan mulai dari kondisi drainase yang kurang baik, panas/suhu udara, air dan hujan, mutu awal produk jalan yang kurang baik serta beban jalan yang berlebihan. Pengawasan dan pengamanan jalan (penanganan muatan lebih) merupakan amanat Undang- undang Nomor 14 Tahun 1992 tentang Lalu Lintas dan Angkutan Jalan. Prasarana jalan yang terbebani oleh volume lalu lintas yang tinggi dan berulang-ulang akan menyebabkan terjadi penurunan kualitas jalan. Sebagai indikatornya dapat diketahui dari kondisi permukaan jalan, baik kondisi struktural maupun fungsionalnya yang mengalami

Kerusakan, Kondisi permukaan jalan dan bagian jalan lainnya perlu dipantau untuk mengetahui kondisi permukaan jalan yang mengalami kerusakan tersebut (Yunardhi, 2018).

Kabupaten Haltim merupakan jalur penghubung antar-kabupaten lainnya di provinsi Maluku Utara, kondisi jalan ruas haltim sebagai jalan lintas kabupaten yang memiliki kondisi dengan tingkat kerusakan yang cukup mengganggu bagi pengendara, beberapa ruas jalan dalam keadaan rusak seperti kerusakan lubang, kerusakan bergelombang, dan juga keretakan pada permukaan jalan, perbaikan jalan di haltim sangat jarang dilakukan karena infrastruktur yang kurang mendukung dan Metode yang digunakan untuk mengidentifikasi kondisi kerusakan jalan masih dilakukan secara manual, membutuhkan waktu dan sumber daya yang signifikan dan juga akurasi yang tidak dapat dipastikan dengan benar karena dapat bersifat subjektivitas dari petugas.

Menurut artikel dari tempo witness Pemerintah Provinsi Maluku Utara telah berencana melakukan perbaikan jalan tersebut sejak 2022. Namun rencana tersebut belum ditindaklanjuti oleh Dinas Pekerjaan Umum dan Penataan Ruang (PUPR),(M.Asrul. 2023).

Convolutional Neural Networks (CNN) merupakan salah satu algoritma dalam pembelajaran mendalam. Tingginya tingkat akurasi dalam proses klasifikasi pada pengolahan citra digital menyebabkan banyak digunakan dalam proses klasifikasi atau identifikasi citra digital (Tempola, 2024).

Dengan adanya penelitian ini untuk mengurangi hambatan dalam pengumpulan dan analisis data kinerja jalan, penggunaan metode secara otomatis dengan memanfaatkan teknologi untuk mendeteksi tingkat kerusakan jalan agar dapat mengidentifikasi masalah kerusakan jalan dengan cepat, deteksi kerusakan jalan yang terjadi pada ruas jalan subaim wasile dengan cara menganalisis kerusakan jalan dan tingkat kerusakannya agar dapat

memberikan alternatif untuk menentukan tingkat kerusakannya serta pemantauan jalan dapat dilakukan secara lebih efektif dan efisien dalam mendukung upaya pemeliharaan dan perbaikan jalan yang berkelanjutan.

Beberapa penelitian dalam mendeteksi dan mengklasifikasi kerusakan jalan telah banyak dilakukan salah satu penelitian yang terkait yaitu Deteksi Kerusakan Jalan Menggunakan Pengolahan *Citra Deep Learning* di Kota Semarang yang dilakukan oleh (Sasmito, 2023). pada penelitian tersebut deteksi kerusakan jalan berbasis *Deep Learning* telah berhasil dilakukan menggunakan model YOLOv4Tiny dengan memanfaatkan *dataset IEEE Road Damage Detection*.

Untuk mendapatkan hasil yang sesuai dilakukan dengan pemanfaatan teknologi *deep learning* Dalam mengklasifikasi tingkat kerusakan jalan dengan menggunakan metode *Convolutional Neural Network* (CNN) untuk pengenalan objek dalam mendapatkan hasil yang akurat (Sasmito, 2023).

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas dapat dirumuskan masalah dalam penelitian ini yaitu Bagaimana teknologi *deep learning*, khususnya CNN, dapat digunakan dalam deteksi tingkat kerusakan jalan?

1.3. Batasan Masalah

Batasan masalah pada penelitian ini yaitu sebagai berikut:

1. Citra jalan yang akan diidentifikasi, adalah jalan rusak yang diambil menggunakan kamera digital pada wilayah Subaim Wasile sepanjang ruas jalan 100-200 m.
2. Metode yang digunakan metode CNN.
3. Jalan yang akan diidentifikasi merupakan jalan kabupaten antar kota.

1.4. Tujuan Penelitian

Berdasarkan hasil dari rumusan masalah yang telah diuraikan oleh penulis, Tujuan dari penelitian ini yaitu bagaimana teknologi *deep learning*, khususnya CNN, dapat diterapkan secara efektif dalam deteksi tingkat kerusakan jalan.

1.5. Manfaat Penelitian

Identifikasi dan deteksi tingkat kerusakan jalan dapat membantu otoritas pemeliharaan jalan untuk mengidentifikasi lokasi kerusakan dengan cepat. Hal ini memungkinkan perbaikan yang lebih efisien dan tepat waktu, mengurangi dampak buruk pada pengguna jalan. sehingga dapat menentukan jalur jalan yang akan dilewati dan juga sebagai alat bantu dalam pemeliharaan dan perbaikan jalan untuk dinas terkait. Juga dapat mengurangi kebutuhan untuk survei lapangan manual yang mahal dan memakan waktu. Ini dapat mengarah pada penghematan biaya dan efisiensi penggunaan sumber daya. serta memberikan manfaat signifikan bagi masyarakat secara keseluruhan.

1.6. Sistematika Penulisan

Sistematis penulisan skripsi ini merupakan pembahasan singkat dari setiap bab yang menjelaskan hubungan antara bab satu dengan bab yang lainnya, yaitu sebagai berikut:

BAB I PENDAHULUAN

Terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematis penulisan.

BAB II TINJAUAN PUSTAKA

Menerapkan teori-teori yang didapat dari sumber-sumber relevan untuk digunakan sebagai panduan dalam penelitian serta menyusun skripsi ini.

BAB III METODE PENELITIAN

Bab ini membahas tentang metode penelitian yang telah dilakukan oleh penulis dengan

permasalahan diangkat.

BAB IV HASIL DAN PEMBAHASAN

Bab ini Menjelaskan hasil yang diperoleh berupa Hasil evaluasi dari model Yang dibuat serta pengujian dalam deteksi kerusakan jalan.

BAB V PENUTUP

Bab ini berisi Kesimpulan dari hasil yang didapatkan serta saran untuk peneliti yang ingin melanjutkan penelitian ini.

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Dalam Penyusunan penyusunan skripsi ini, penulis terinspirasi dan mereferensi dari penelitian-penelitian sebelumnya yang terkait dengan latar belakang masalah pada skripsi ini. Adapun penelitian yang berhubungan dengan skripsi ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terkait.

No	Nama dan Tahun	Judul	Hasil Penelitian
1.	Nur Fadila, Rifki Kosasih (2019)	Klasifikasi jenis Kendaraan Menggunakan Metode <i>Convolutional Neural Network</i> (CNN)	Pada Penelitian ini klasifikasi kendaraan dengan metode <i>Convolutional Neural Network</i> (CNN) telah berhasil dilakukan dengan tingkat akurasi sebesar 73,33%. Model CNN yang digunakan terdiri dari empat lapisan konvolusi dengan ukuran filter 3×3, fungsi aktivasi yang digunakan yaitu reLu, dan 2 pooling layer dengan ukuran 2×2.
2.	W mellysa, AF Dewi, M Misrina, S suryanti, R Rachmawati (2022)	Pengaruh Algoritma Deep Learning dalam Meningkatkan Akurasi Sistem Pendeteksian Kondisi Jalan Raya	Dari hasil percobaan pada data testing dan evaluasi ditemukan bahwa dengan menggunakan VGG16 berhasil mendeteksi kerusakan pada image yang diberikan. Dengan basis deep learning dan menggunakan CNN untuk ekstraksi fitur image, akan sangat membantu proses training dan pntuan kelas input image. Model ini berhasil menghasilkan akurasi di angka 97.3 % dan error deteksi sebesar 2.7%
3	Stephen Ekaputra Limantoro, Yosi Kristian, Devi Dwi Purwanto.(2018)	Pemanfaatan <i>Deep Learning</i> pada Video Dash Cam untuk Deteksi Pengendara Sepeda Motor	Berdasarkan hasil eksperimen, deteksi pengendara sepeda motor dengan menggunakan CNN mampu menangani fitur yang kompleks dan variatif. Proses pelatihan sebagian besar menggunakan arsitektur

			dan parameter dari penelitian sebelumnya.
4	Agung wira Mulia,ikhwan Ruslianto, Dwi Marisa Midyanti,(2023)	Klasifikasi kerusakan jalan pada citra jalan raya pontianak dan sekitarnya dengan menggunakan <i>Convolutional Neural Networks</i> (CNN)	Aplikasi yang telah dibangun mampu memberikan kelas dari kerusakan jalan dan mengklasifikasikan dengan baik. Dalam proses pengolahan citra sebelum di lakukan pelatihan adalah dengan mengubah ukuran citra menjadi citra 32x32 serta mengubah citra menjadi grayscale. Pada pengujian epoch ke 6 ,akurasi yang didapatkan adalah 80%.Pada pengujian epoch ke 6 ,akurasi yang didapatkan adalah 80%. Parameter yang digunakan dalam. Berdasarkan pada arsitektur LeNet-5. Pada tahap pengujian data uji yang
5.	Yoga Triardhana , Bandi Sasmito, Firman Hadi,(2021)	Identifikasi Kerusakan Jalan Menggunakan <i>Metode Deep Learning</i> (DI) Model <i>Convolutional Neural Networks</i> (CNN)	Hasil titik persebaran kerusakan jalan jika dibandingkan dengan validasi manual menggunakan GPS smartphone memiliki nilai RMSE sebesar 7,82 meter. Nilai akurasi model yang didapat dengan membandingkan hasil dari model dan dari data hasil validasi dengan menggunakan confusion matrixpada setiap kelas kerusakan jalan memiliki nilai <i>overallaccuracy</i> sebesar 85,34% dan nilai kappa <i>accuracy</i> sebesar 82,36%
6.	Muhammadun Basyarul Habibi1, Setyoningsih Wibowo,(2023)	Pengembangan Aplikasi Deteksi Kerusakan Lubang Jalan Berbasis Android	Dalam penelitian ini, dikembangkan sebuah aplikasi berbasis Android yang mengintegrasikan model prediksi kerusakan lubang jalan. Aplikasi ini memungkinkan pengguna untuk mengakses informasi prediksi, mengunggah foto lubang jalan rusak, dan mendapatkan prediksi yang akurat. Analisis kebutuhan mengidentifikasi

			beberapa kebutuhan pengguna, termasuk akses informasi prediksi, antarmuka yang intuitif, akurasi tinggi, kemudahan pengunggahan foto, dan integrasi dengan fitur "get current location" pada perangkat Android
7.	Pulung Adi Nugroho, Indah Fenriana. Rudy Arijanto,M.Kom,(2020)	Implementasi Deep Learning Menggunakan <i>Convolutional Neural network</i> (CNN) Pada Ekspresi Manusia	Berdasarkan penelitian dan hasil penerapan metode Convolutional Neural Network (CNN) dalam mengklasifikasikan ekspresi wajah, Metode Convolutional Neural Network (CNN) sangat cocok digunakan untuk menguji sebuah citra, karena prosesnya yang berlapis-lapis, terbukti dengan 35 citra, 28 citra bisa ditebak dengan benar walaupun ekspresi hanya berbeda tipis- tipis.
8.	Andriyana Fajar, Dadang Iskandar Mulyana, Aziz Septian Amrullah, Kurniawan Irfan Nauval,(2022)	Identifikasi Kerusakan Jalan DENGAN Metode Faster R- Cnn Studi Kasus DI Jalan Pakansari Bogor Jawa Barat	Analisis akurasi model yang dihasilkan menggunakan matriks konfusi dengan sampel titik di sepanjang jalur yang diamati. Menurut rumus Slovin, jumlah sampel yang digunakan 10,00 km, dengan tingkat kepercayaan 90%, 116 sampel dihasilkan dari Pakansari-Bogor, Jawa Barat.[10] Analisis akurasi spasial dilakukan dengan membandingkan lokasi spasial hasil model CNN dengan data validasi [11]. Data otentikasi diperoleh dengan menggunakan smartphone untuk video dan data lokasi.[12] Kerusakan jalan diperoleh dengan menggunakan alat bantu visual secara manual oleh penulis
9.	Rosyady, Aditya aphisca Rahani, Faisal Fajri Baswara, Ahmad Raditya Cahaya(2021)	Amarto (Damaged Road Detector): Purwarupa Sistem Deteksi Dan Analisator Kerusakan Jalan	Sistem ini mampu menganalisa dan menghitung luas jalan berlubang dengan tingkat error 0.358% - 4.19%, namun nilai parameter lower, upper HSV, gaussian blur kernel, alpha, beta

		Raya Kota Yogyakarta Berbasis Citra Digital Dan Gps Tim	pada fungsi masih beragam. Program yang berperan penting dalam proses pendeteksian jalan berlubang adalah equalizer histogram
10.	Firman Tempola, Retantyo Wardoyo Aina Musdholifah, Rosihan, Lilik Sumaryanti (2024)	Klasifikasi jenis cengkeh menggunakan algoritma convolution neural network dengan optimasi hyperparamter	Klasifikasi jenis cengkeh dengan menerapkan metode CNN dapat mengidentifikasi jenis cengkeh dengan baik. Akurasi tertinggi diperoleh yaitu 84% berdasarkan hasil pengujian. Kecepatan pembelajaran yang diterapkan adalah 0,001, 50 zaman, dan 16 ukuran batch. Untuk eksperimen menggunakan beberapa model pembelajaran transfer seperti VGG16, Resnet50, DenseNet151, MobileNetV2, InceptionV3, dan arsitektur VGG16 yang dimodifikasi, akurasi setiap model berbeda-beda. Namun akurasi tertinggi diperoleh ketika arsitektur VGG16 dimodifikasi yaitu 99,11%. Memodifikasi arsitektur VGG16 sangat berpengaruh dalam klasifikasi khususnya pada data cengkeh
	usulan	Pada penelitian sebelumnya metode menggunakan metode Convolutional Neural Network (CNN). Perbedaan mungkin terletak pada arsitektur CNN, fungsi aktivasi, dan parameter lainnya Penggunaan metode Convolutional Neural Network (CNN). dalam penelitian sebelumnya ini direkomendasikan untuk inovasi dalam pengamatan kerusakan jalan karena mampu dilakukan secara otomatis tanpa bantuan manusia. cepat dalam pengolahannya serta memiliki akurasi yang baik,	

Berikut ini merupakan beberapa penjelasan untuk membedakan antara penelitian sebelumnya dengan dan penelitian yang saya teliti, diantaranya sebagai berikut:

1. pendekatan Metode:

judul yang saya teliti menggunakan metode CNN. Perbedaan mungkin terletak pada

arsitektur CNN, fungsi aktivasi, dan parameter lainnya. sementara yang lain mungkin menggunakan algoritma *deep learning* lainnya seperti Faster R-CNN.

Ada beberapa penelitian yang fokus pada pengembangan aplikasi berbasis Android, sementara yang lain mungkin fokus pada analisis menggunakan citra digital dan GPS.

2. Subjek Studi Kasus

Penelitian Identifikasi Kerusakan Jalan: Studi Kasus Pada Wasile (merupakan suatu kecamatan pada kabupaten haltim).

3. Algoritma dan teknik

Judul yang sedang diteliti menggunakan CNN, berbeda dengan penelitian sebelumnya, penelitian sebelumnya menggunakan algoritma deteksi atau identifikasi yang berbeda, pada penelitian sebelumnya menggunakan algoritma lain seperti *Faster Faster R-CNN* atau YOLO (*You Only Look Once*). Perbedaan dalam algoritma dapat mempengaruhi kinerja sistem dalam mendeteksi atau mengidentifikasi kerusakan jalan dengan cara yang lebih akurat, efisien, atau cepat.

4. Hasil dan Tingkat Akurasi

Pada penelitian sebelumnya Penggunaan metode dalam penelitian ini direkomendasikan untuk inovasi dalam pengamatan kerusakan jalan karena mampu dilakukan secara otomatis tanpa bantuan manusia. cepat dalam pengolahannya serta memiliki akurasi yang baik.

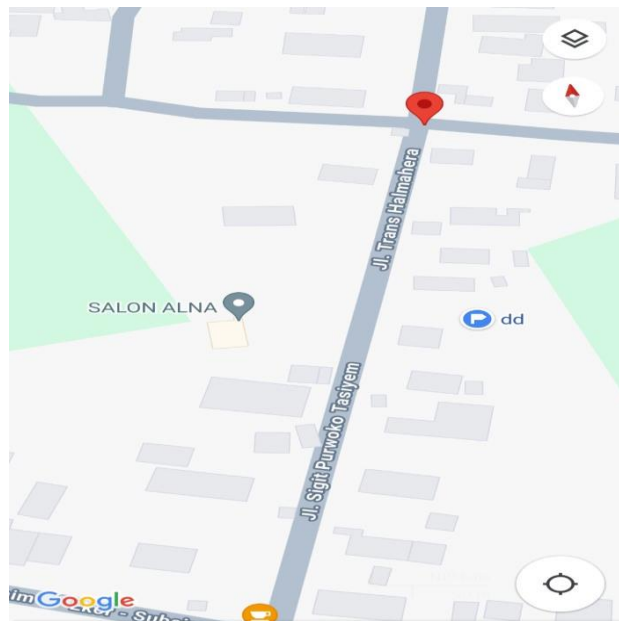
2.2. Pengertian Jalan

Menurut Peraturan Pemerintah Nomor 34 Tahun 2006, jalan adalah prasarana transportasi darat yang meliputi segala bagian jalan, termasuk bangunan pelengkap dan

perlengkapannya yang diperuntukkan bagi lalu lintas, yang berada pada permukaan tanah, di atas permukaan tanah, dibawah permukaan tanah dan air, serta diatas permukaan air, kecuali jalan kereta api, jalan lori, dan jalan kabel (Yunardhi, 2018).

2.3. Ruas Jalan Subaim

Kecamatan Wasile adalah salah satu kecamatan di Kabupaten Halmahera Timur, Provinsi Maluku Utara, Indonesia. jalan-jalan utama biasanya menghubungkan pusat kecamatan dengan desa-desa di sekitarnya, serta menghubungkan Kecamatan Wasile dengan kecamatan lainnya dan pusat kabupaten Beberapa jalan mungkin memerlukan perbaikan atau peningkatan untuk meningkatkan aksesibilitas dan keselamatan pengendara. Berikut dapat dilihat pada gambar 2.1.



Gambar 2.1. ruas jalan halmim (google Maps, 2023).

2.4. Kerusakan Jalan

Kondisi jalan yang di lalui oleh volume lalu lintas yang tinggi dan berulang-ulang akan mempengaruhi kondisi kontruksi jalan, dan mengakibatkan terjadinya penurunan kualitas jalan tersebut, sehingga berdampak pada keamanan, kenyamanan, dan kelancaran dalam

berlalu lintas (Aptarila, 2020). Kerusakan jalan merupakan suatu kerusakan pada lapis perkerasan jalan yang dibagi menjadi dua macam yaitu:

1. Kerusakan Struktural Kerusakan jenis ini merupakan kerusakan yang terjadi pada struktur jalan sehingga jalan tidak mampu lagi menopang beban lalu lintas yang lewat di atasnya. Perlu dilakukan perbaikan untuk memperkuat struktur jalan supaya dapat menopang beban lalu lintas.
2. Kerusakan Fungsional Kerusakan jenis ini terdapat pada permukaan jalan yang membuat fungsi jalan tidak optimal. Perkerasan jalan masih dapat menopang beban lalu lintas di atasnya tetapi kenyamanan pengguna lalu lintas menjadi terganggu (Yoga Triardhana, 2020).

2.5. Jenis-Jenis Kerusakan Jalan

Terdapat beberapa jenis-jenis kerusakan jalan yang sering terjadi, berikut terkait dengan kerusakan jalan sebagai berikut: (Pramestya, 2018).

1. Retak Kulit Buaya (*Alligator Cracking*)

Retak jenis ini berbentuk sebuah jaring-jaring yang menyerupai kulit buaya, dengan lebar jaring lebih besar atau sama dengan 3 mm. Retak jenis ini disebabkan oleh beban lalu lintas yang terlalu berat dan terus menerus. Dapat dilihat pada Gambar 2.2.



Gambar 2.2. *Alligator Cracking* (Pramestya, 2018).

2. Keriting (*Corrugation*)

Bentuk kerusakan ini berupa gelombang pada permukaan jalan yang arahnya melintang jalan. Kerusakan jenis ini terjadi akibat pengereman kendaraan. Salah satu penyebabnya yaitu stabilitas lapis permukaan yang rendah, terlalu banyak menggunakan agregat halus dan pondasi yang bergelombang. Dapat dilihat pada Gambar 2.3.



Gambar 2.3. *Corrugation* (Prametsya, 2018).

3. Ambblas (*Depression*)

Bentuk kerusakan jenis ini berupa turunnya permukaan lapisan pada lokasi tertentu dengan atau tanpa retak. Umumnya, kedalaman ambblas lebih dari 2 cm dan akan menampung atau meresap air. Penyebabnya adalah beban berat kendaraan yang berlebihan, sehingga struktur bagian bawah tidak mampu menahan beban, penurunan bagian perkerasan dikarenakan oleh turunnya tanah dasar, dan pelaksanaan pemadatan yang kurang baik. Dapat dilihat pada Gambar 2.4.



Gambar 2.4. *Depression* (Prametsya, 2018).

4. Lubang (*Potholes*)

Kerusakan jenis ini berbentuk seperti mangkok dan dapat menampung dan meresap air pada bahu jalan. Kerusakan ini terkadang terjadi dekat retakan, atau di daerah yang sering tergenang oleh air. Kemungkinan penyebabnya adalah seperti kadar aspal rendah, sehingga agregatnya mudah terlepas atau lapis permukaannya tipis, pelapukan aspal, dan suhu campuran tidak memenuhi syarat. Dapat dilihat pada Gambar 2.5.

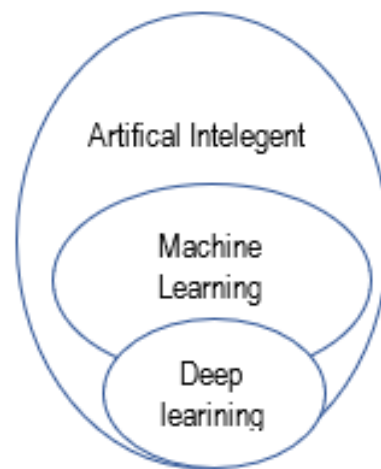


Gambar 2.5. Lubang (Prametsya, 2018).

2.6. **Deep Learning**

Deep Learning adalah bagian dari kecerdasan buatan dan *machine learning* yang merupakan pengembangan dari *neural network multiple layer* untuk memberikan ketepatan tugas seperti deteksi objek, pengenalan suara, terjemahan bahasa dan lain-lain. *Deep Learning* berbeda dari teknik machine learning yang tradisional, karena *deep learning* secara otomatis melakukan representasi dari data seperti gambar, video atau text tanpa memperkenalkan aturan kode atau pengetahuan domain manusia (Raup, 2022).

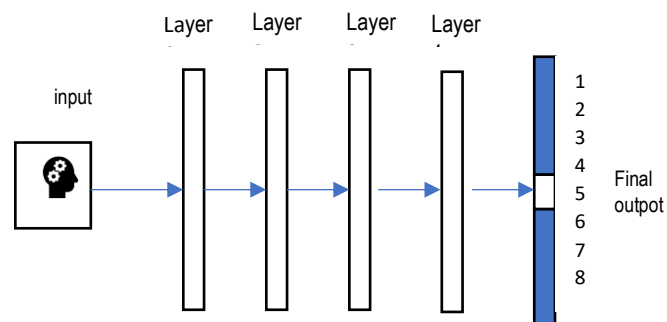
deep learning (pembelajaran mendalam) adalah sub bidang yang spesifik dari *machine learning* (pembelajaran mesin). Kata '*deep*' pada *deep Learning* tidak mengacu pada pemahaman yang lebih dalam untuk pendekatan, sebaliknya ini merupakan gagasan dari *layer* yang berurutan. Dapat dilihat pada Gambar 2.6.



Gambar 2.6. Hubungan *Deep Learning* Dengan *Machine Learning* (Raup, 2022).

Banyak lapisan yang digunakan pada model data disebut kedalaman ('*depth*') dari model. *Deep learning* moderen sering melibatkan puluhan atau bahkan ratusan layers yang berurutan terobosan yang telah dicapai. Beberapa metode dari *deep learning* adalah *Long Short Term Memory* (LSTM) dan CNN (Pramestya, 2018).

Ditinjau dari kegunaannya untuk analisis citra, *deep learning* dapat digunakan untuk klasifikasi gambar (*single object*), mengetahui lokasi objek (hasil berupa *bounding box* objek yang terdapat pada citra, serta identifikasi objek (mengetahui lokasi dan memberi nama kelas objek tersebut). Tingkatan selanjutnya dari semua itu adalah segmentasi objek dimana objek yang diidentifikasi akan diberikan blok pada piksel objek tersebut (Yoga Triardhana, 2020). Dapat dilihat pada Gambar 2.7.

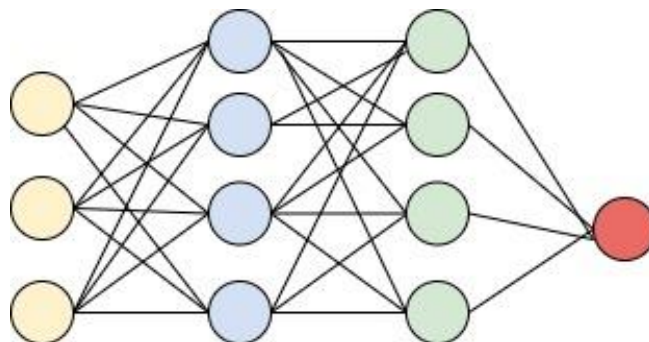


Gambar 2.7. Klasifikasi Dengan *Deep Learning* (Yoga Triardhana, 2020).

2.7. Neural Network

Neural Network (NN) merupakan metode dengan sistem kerja meniru fungsi otak manusia. Dalam otak manusia diyakini terdiri dari jutaan neuron atau unit pengolahan kecil yang sistem kerjanya *paralel*. *Neuron* saling terhubung satu sama lain melalui koneksi *neuron*. *Neural network* dapat dianggap sebagai sistem kotak hitam yang menerima masukan dari lingkungan dan menghasilkan keluaran. Elemen pemrosesan dan pembobotan terkandung dalam *neural network*, elemen tersebut saling terhubung. Setiap susunan dalam jaringan berisi oleh kelompok elemen pemrosesan, (Hadianto, 2019). Ada beberapa istilah yang akan kita temui dalam *Neural Network*. Dapat dilihat pada gambar 2.8.

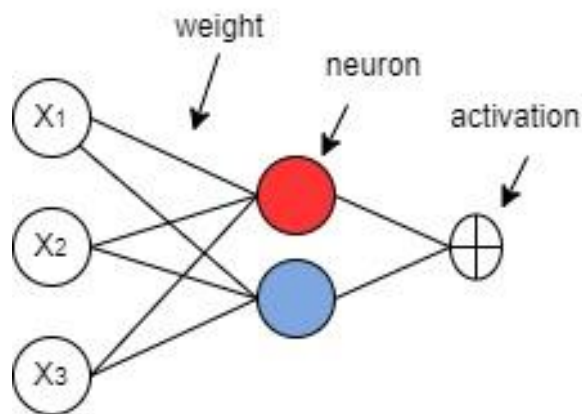
1. *Input layer, hidden layer, output layer*.
2. *Weight* atau bobot.
3. Node atau *neuron*.
4. Fungsi Aktivasi



Gambar 2. 8. Arsitektur *Neural Network* (Hadianto, 2019).

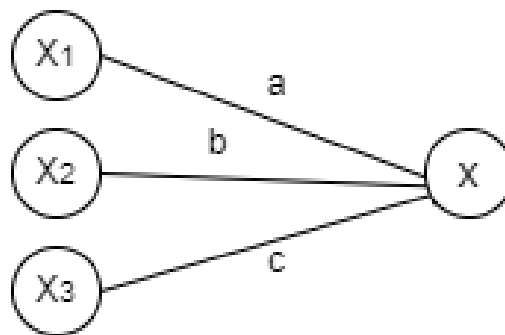
Setiap elemen pemrosesan mengumpulkan nilai dari semua *input* yang terhubung ke elemen pemrosesan dan menghasilkan *output* melalui operasi. Dari gambar diatas bisa dilihat bahwa ada tiga susunan yang membangun *neural network*, yaitu *input layer, hidden layer*, dan *output layer*. *Artificial Neural Network* (Jaringan Syaraf Tiruan) merupakan bentuk *non-linear* yang kompleks, dibentuk dari elemen yang secara individu mempunyai karakter

mirip dengan model regresi. Jaringan syaraf tiruan dapat digambarkan dengan grafik, dan beberapa sub-grafik yang memiliki karakter yang sama dengan grafik sebelumnya. Representasi *neural network* dalam gambar dapat dilihat pada gambar 2.9.



Gambar 2. 9. Representasi *Neural Network* (Hadianto, 2019).

Adapun rumus atau dan interpretasi dalam node yang digunakan dalam *neural network* dapat dilihat pada gambar 2.10.



Gambar 2. 10. Rumus dan Interpretasi *neural network* (Hadianto, 2019).

Rumus dan Interpretasi *neural network* dapat dilihat pada persamaan 2.1.

$$y = \sigma(ax^1 + bx^2 + cx^3 \dots \dots \dots (2.1)$$

Fungsi aktivasi sangat umum digunakan dalam *neural network*. Alasan utama menggunakan fungsi aktivasi adalah agar *neural network* mengenali data yang non-linear, karena *output* yang dihasilkan dari *neural network* jarang sekali bersifat linear. (Muhamad, 2020). Fungsi aktivasi *neural network* atau dapat dikenal dengan *Logistic Regression* merupakan salah

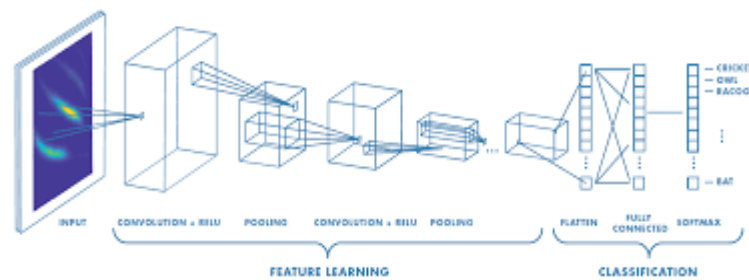
satu jenis Regresi Linier yang menggunakan Fungsi Aktivasi Sigmoid yang merupakan salah satu jenis Fungsi Aktivasi Nonlinier. Berikut rumus dari fungsi Sigmoid dapat dilihat pada persamaan 2.2.

$$\sigma(x) = \frac{1}{1+e^{-x}} \dots \dots \dots (2.2)$$

Pada persamaan 2.2. nilai a, b, dan c merupakan nilai bobot. Bobot adalah nilai matematis dari sebuah koneksi antar neuron, misalnya bobot pada *neuron input* ke *hidden layer*, dan bobot pada *hidden layer* ke *output*. Nilai bobot dan bias diberikan dengan cara melakukan pembangkitan nilai acak dengan interval sembarang. Nilai acak bisa dibangkitkan dalam interval $[-1, +1]$ atau $[-0.5, +0.5]$. Beberapa penelitian yang telah dilakukan mengisyaratkan agar pada awal pembangkitan bilangan random dilakukan dengan nilai interval yang kecil (Santoso, 2019).

2.8. **Convolutional Neural Network (CNN)**

CNN adalah pengembangan dari *Multilayer Perceptron* (MLP) yang termasuk dalam *neural network* bertipe *feed forward* (bukan berulang). CNN adalah *neural network* yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. CNN digunakan untuk menganalisis gambar *visual*, mendeteksi dan mengenali objek pada *image*, yang merupakan vektor berdimensi tinggi yang akan melibatkan banyak parameter untuk mencirikan jaringan. Secara garis besar, CNN tidak terlalu jauh berbeda dengan *neural network* biasanya. CNN terdiri dari *neuron* yang memiliki *weight*, *bias*, dan *activation function*, arsitektur Algoritma CNN dapat dilihat pada gambar 2.11. (Nugroho, 2020).



Gambar 2. 11. Arsitektur Algoritma CNN (Nugroho, 2020).

2.8.1. *Input layer*

Dalam konteks CNN, lapisan *input* (*input layer*) berfungsi sebagai penerima data mentah yang biasanya berupa gambar. Fungsi dan karakteristik lapisan *input* dalam CNN memiliki beberapa perbedaan dan persamaan dengan jaringan saraf tiruan biasa dimana dapat menampung nilai piksel dari citra sebagai masukan. Buat citra dengan dimensi 64x64 dengan 3 *channel* warna, RGB (*Red, Green, Blue*) sehingga yang jadi masukan ialah piksel array yang berdimensi 64x64x3 contoh dari *input layer* RGB dapat dilihat pada gambar 2.12. (Sofyan, 2019).



Gambar 2. 12 *Input Layer* RGB (Sofyan, 2019).

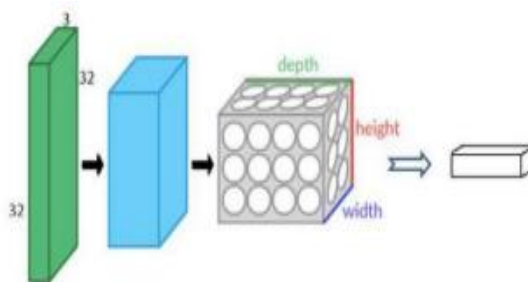
terdapat tiga matriks dalam RGB yaitu matriks warna *red*, *green*, dan *blue*. Pada setiap *pixel* warna memiliki rentang nilai intensitas antara 0 sampai 255, rumus normalisasi RGB pada Persamaan 2.3.

$$R = \frac{R}{255} \quad G = \frac{G}{255} \quad B = \frac{B}{255} \dots \dots \dots (2.3)$$

r merupakan komponen normalisasi dari nilai R (*red*), g adalah komponen normalisasi dari nilai G (*green*), dan b adalah komponen normalisasi dari nilai B (*blue*). Normalisasi nilai RGB penting dalam pemrosesan citra Sehingga kita dapat memastikan bahwa setiap nilai komponen warna berada dalam rentang yang seragam, yang memudahkan berbagai proses pemrosesan citra dan analisis lebih lanjut.

Arsitektur CNN terdiri dari lapisan *input (input layer)*, lapisan *output (output layer)* dan beberapa lapisan tersembunyi (*hidden layers*). Lapisan tersembunyi terdiri dari *convolutional layers*, *ReLU layer*, *fully connected layers* dan *loss layer*.

arsitektur CNN merupakan arsitektur tiga dimensi yaitu : lebar (*width*), tinggi (*height*) dan dalam (*depth*) seperti yang digambarkan pada Gambar 2.13. Bagian lebar dan tinggi menyatakan dimensi dari citra, sedangkan bagian dalam (*depth*) merupakan kanal *Red*, *Green* dan *Blue* (Zahra, 2023).dapat dilihat pada gambar 2.13.



Gambar 2. 13. Arsitektur tiga dimensi CNN (Zahra, 2023).

Secara garis besar CNN memiliki 2 tahap utama yaitu *feature learning* dan *classification*, yang di setiap bagian CNN terdapat ada 2 proses utama yaitu *feed- forward* dan *backpropagation*. CNN untuk *tahap feature learning* terdapat *convolutional layer*, *ReLU* (fungsi aktivasi) dan *pooling layer*. CNN untuk tahap *classification* terdapat *flatten*, *fully-connected layer*, dan *loss layer* (Zahra, 2023).

2.8.2. Feature Learning

Pada *feature learning* terdapat beberapa proses atau tahapan yang terjadi yaitu sebagai berikut:

1. Convolutional layer

Lapisan konvolusi (*Convolutional Layer*) adalah komponen utama dalam arsitektur CNN yang bertanggung jawab untuk mengekstraksi fitur dari *input* data, biasanya berupa gambar. penjelasan rinci mengenai lapisan konvolusi *Convolutional layer* merupakan operasi matematis yang terdapat dalam CNN dimana *filter* atau kernel ini diterapkan pada inputan gambar, *filter* atau kernel ini biasanya berukuran matrik kecil yang akan melintasi pada seluruh data input untuk melakukan operasi titik per titik, layer atau lapisan citra yang dimiliki oleh citra yang telah diinputkan. *Input* citra yang memiliki ukuran $n \times n$ akan dipisahkan sehingga menghasilkan 3 *channel*, yaitu *layer Red*, *layer Green* dan *layer Blue*, sehingga terbentuk multidimensional *array* dengan ukuran $n \times n \times 3$. *Convolutional layer* memiliki susunan *neuron* yang membentuk filter dengan membentuk *matriks*, dimana *matriks* ini memiliki panjang dan lebar dengan nilai tertentu. Filter dengan *matriks* ini selanjutnya akan dikomputasi dengan 3 *layer* citra input yang sebelumnya telah dipisahkan dengan RGB. Berikutnya komputasi yang telah dilakukan ini akan menghasilkan *output matriks* yang baru dan disebut dengan *layer output*. Proses konvolusi ini dilakukan sebanyak ratusan bahkan ribuan kali tergantung dengan *hyperparameter* yang dipakai (Saputra, 2022).

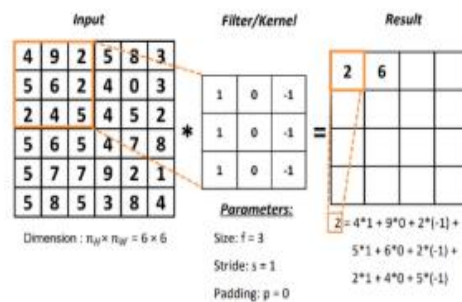
a. Filter/kernel

Operasi *convolution Citra* dengan sebuah filter atau kernel seperti yang digambarkan terdapat 1 *channel input* dengan ukuran matriks 6×6 dengan ukuran matriks dari filter yaitu

3×3 dan menghasilkan *output matriks* hasil operasi *convolution* dengan ukuran 4×4 . Dapat dilihat pada Gambar 2.14.

Rumus saat menghitung proses *convolution* sebagai berikut:

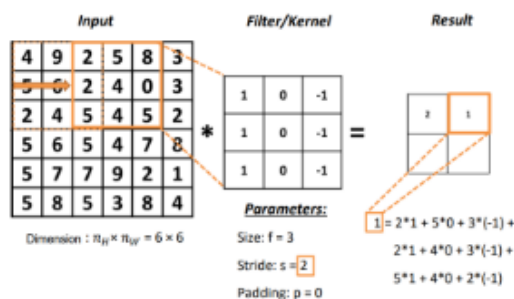
$$a.b = \sum_{i=1}^n a_i b_i + a_2 b_2 + \dots + a_n b_n \dots \dots \dots (2.4)$$



Gambar 2. 14. Operasi *convolution* citra menggunakan sebuah filter atau kernel (Saputra, 2022).

b. *Stride*

Stride merupakan parameter yang ditentukan guna untuk melihat pergeseran yang terjadi pada piksel. Pergeseran piksel ini dilakukan secara *horizontal* dan *vertikal*. Sebagai contoh jika *stride* bernilai 2 maka akan bergeser sebanyak 2 piksel secara *horizontal* dan *vertikal* (Saputra, 2022) seperti yang digambarkan pada Gambar 2.15.

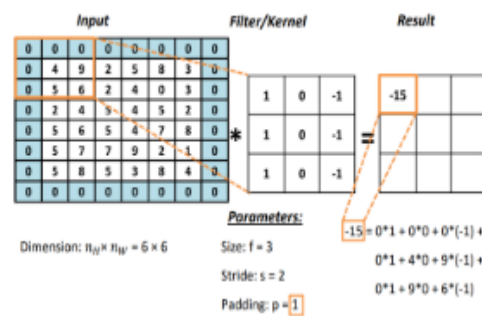


Gambar 2. 15. Pengaruh *stride* terhadap operasi *convolution* citra (Saputra, 2022).

c. *Padding*

Teknik *padding* merupakan teknik yang dilakukan guna menambahkan nilai piksel

pada setiap pinggiran pada *input layer*, Penambahan *padding* ini dapat menjaga informasi di setiap tepi agar tidak hilang karena setelah terjadi perhitungan *konvolusi*. Penambahan *padding* seperti yang digambarkan pada Gambar 9 dengan matriks berdimensi 6×6 dengan penambahan 1 *padding* sehingga menjadi *matriks* berdimensi 8×8 dan stride 2 akan berubah menjadi matriks 3×3 , jika dibandingkan dengan Gambar 8 yang memiliki dimensi matriks yang sama dan stride 2 tanpa penambahan *padding* menghasilkan *matriks* dengan ukuran 2×2 sehingga piksel yang berada di pinggir *input layer* hilang beserta informasinya. (Saputra, 2022). Dapat dilihat pada Gambar 2.16.



Gambar 2. 16. Penambahan *padding* terhadap operasi *convolution* (Saputra, 2022).

2. Pooling layer

Poling Layer merupakan proses yang bertujuan untuk mereduksi dimensi hasil dari *feature map*, karena nilai piksel tersebut tidak semuanya digunakan dan berharga. *Pooling layer* berfungsi untuk mempercepat proses komputasi tanpa harus kehilangan *piksel-piksel* yang memiliki informasi yang berguna. Terdapat 2 jenis *pooling layer* yaitu *max pooling* untuk mencari nilai tertinggi dari piksel- piksel tersebut dan *average pooling* untuk mencari nilai rata-rata dari piksel tersebut seperti yang digambarkan. (Saputra, 2022).

Lapisan *pooling* yang dimasukkan di antara lapisan konvolusi secara berturut-turut pada susunan 15 arsitektur CNN secara progresif bisa mengurangi ukuran dari volume

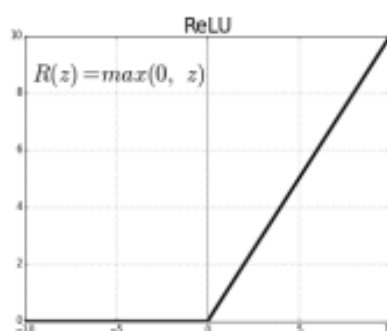
output pada *Feature Map*. *Pooling layer* membuat data menjadi lebih kecil, lebih mudah dikelola, dan lebih mudah untuk mengontrol *overfitting*, dapat dilihat pada Gambar 2.17.



Gambar 2. 17. *Max pooling* dan *average pooling* (Saputra, 2022).

3. *Rectified Linear Units* (ReLU)

Layer Rectified Linear Units (ReLU) layer yang merupakan fungsi aktivasi $f(x) = \max(0, x)$. *ReLU* layer dapat meningkatkan sifat non linearitas dari fungsi keputusan dan jaringan yang ada secara keseluruhan tanpa memberikan pengaruh pada bagian reseptif yang terdapat pada *convolution layer* (Suyanto, 2019). Fungsi aktivasi *ReLU* juga dapat mempercepat proses pada training data (Primartha, 2018). Grafik *ReLU* digambarkan pada gambar 2.18.



Gambar 2. 18 Fungsi Aktivasi *Rectified Linear Units* (ReLU) (Primartha, 2018).

2.8.3. **Classification**

Klasifikasi adalah proses dalam pembelajaran mesin dan ilmu data untuk mengidentifikasi kategori atau kelas dari objek atau data tertentu berdasarkan fitur-fitur yang

dimilikinya. Dalam konteks pembelajaran mesin, klasifikasi adalah tipe masalah yang tujuannya adalah untuk memetakan input kedalam salah satu dari beberapa kategori yang telah ditentukan sebelumnya.

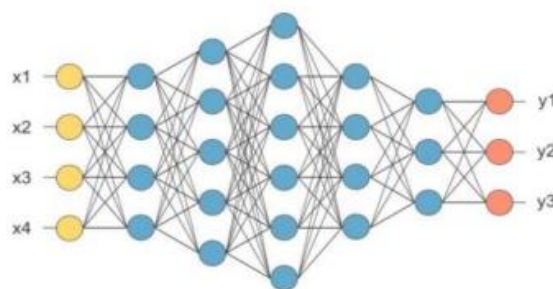
Pada *classification* terdapat beberapa proses atau tahapan yang terjadi yaitu sebagai berikut:

1. *Flattening*

merupakan proses mengubah data dari bentuk multi-dimensi (seperti matriks atau tensor) menjadi bentuk satu dimensi. Dalam konteks jaringan saraf tiruan, flattening digunakan untuk mengubah keluaran dari lapisan konvolusi (yang biasanya berbentuk 2D atau 3D) menjadi vektor satu dimensi sebelum diteruskan ke lapisan dense (*fully connected*).

2. *Fully connected layer Fully*

Fully connected layer merupakan proses *multilayer perceptron* (MLP), yang merupakan algoritma yang termasuk ke neural network yang di dalamnya terdapat *input layer*, *neuron* sebagai *hidden layer*, *activation function*, *output layer* dan *loss function*. Proses selanjutnya adalah evaluasi dari hasil prediksi apakah hasil ini mendekati hasil sebenarnya atau tidak. Jika hasil evaluasi ini masih jauh dari hasil sebenarnya, maka perlu dilakukan kembali *backpropagation* dan melakukan *update* setiap bobot pada *hidden layer*. Dapat dilihat pada gambar 2.19. (Saputra, 2022).



Gambar 2. 19 *Fully Connected Layer* (Saputra, 2022).

3. *Loss Layer*

Loss layer merupakan lapisan terakhir dalam CNN yang berguna untuk menentukan pelatihan dalam memberikan penalti atau hasil akhir terhadap penyimpangan antara hasil prediksi dan label. Terdapat beberapa *loss function* yaitu (Suyanto, 2019):

- a. *Softmax loss* untuk memprediksi satu dari beberapa kelas yang saling eksklusif. Fungsi Softmax digunakan pada multiclass classification, jumlah nilai probabilitasnya adalah 1, nilai yang tinggi akan memberikan nilai probabilitas yang tinggi pula jika dibandingkan dengan nilai yang lainnya.
- b. *Sigmoid cross-entropy loss* untuk memprediksi nilai probabilitas dalam interval $[0, 1]$.
- c. *Euclidean loss* untuk regresi nilai *continue*.

2.9. Fitmodel

Istilah yang merujuk pada proses pelatihan model dalam konteks pembelajaran mesin dan statistik. Secara khusus, dalam konteks *deep learning* dan penggunaan CNN, fitmodel adalah proses di mana model pembelajaran mesin atau *deep learning* menyesuaikan parameter internalnya berdasarkan data pelatihan yang diberikan. Tujuannya adalah untuk meminimalkan kesalahan prediksi dengan mengoptimalkan fungsi kerugian (*loss function*) sehingga model dapat membuat prediksi yang akurat pada data yang tidak terlihat sebelumnya.

2.10. Evaluasi model

Evaluasi model merupakan tahapan yang penting dalam *deep learning*. Dalam konteks penelitian ini, hal tersebut berkaitan dengan pengetahuan seberapa baik performa model tersebut dalam klasifikasi. Salah satu perangkat evaluasi yang umum digunakan untuk klasifikasi adalah *confusion matrix* atau disebut juga tabel *confusion* merupakan

sebuah tabel yang menggambarkan prediksi dan label sebenarnya dari sebuah klasifikasi. Contoh ilustrasi *confusion matrix* untuk permasalahan klasifikasi biner (Hu, 2020) ditunjukkan pada tabel 2.2.

Tabel 2.2. *confusion matrix* (hu, 2020).

	P'(predicted)	N'(predicted)
P(actual)	True Positive (TP)	False Negative (FN)
N(actual)	False Negative (FN)	True Positive (TP)

Pada Tabel 2.1 tersebut bisa diketahui parameter akurasi, presisi, *recall* dan *f score*.

Berikut adalah penjelasan dan rumus untuk masing-masing parameter tersebut:

1. *Accuracy* adalah perbandingan antara jumlah prediksi yang benar dengan jumlah keseluruhan prediksi benar dan salah (Hu, 2020). Berdasarkan tabel di atas maka akurasi dapat dirumuskan, dapat dilihat pada persamaan 2.1.

$$accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \dots \dots \dots (2.5)$$

2. *Precision* adalah perbandingan antara prediksi benar terhadap sebuah label dengan keseluruhan prediksi benar dan salah terhadap label tersebut, atau jumlah perbandingan *True Positive* terhadap *Predicted Positive*, Presisi dapat diwakili dengan rumus, dapat dilihat pada persamaan 2.2.

$$precision = \frac{TP}{(TP+FP)} \dots \dots \dots (2.6)$$

3. *Recal* atau bisa disebut juga dengan sensitivity mengukur seberapa baik model dalam menghindari *false negative*, atau jumlah perbandingan *True Positive* terhadap *Acual Positive*, *Recall* dapat dihitung dengan rumus, dapat dilihat pada persamaan 2.3.

$$reccal = \frac{TP}{(TP+FN)} \dots \dots \dots (2.7)$$

4. *F1-score* adalah kombinasi rata-rata dari presisi dan *recall*, *F1-score* bisa dihitung

dengan menggunakan rumus, dapat dilihat pada persamaan 2.4.

$$F1 - Score = \frac{2TP}{(2TP+FP+FN)} \dots\dots\dots (2.8)$$

2.11. *Python*

Python adalah bahasa pemrograman *interpretatif* multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. *Python* juga didukung oleh komunitas yang besar, selain itu *Python* juga memiliki *Library* yang lengkap dan memiliki dukungan komunitas yang kuat karena *Python* bersifat *open source*. Untuk menuliskan *source code Python* anda dapat menggunakan IDE seperti *vs code*, *sublime text*, *PyCharm* atau anda juga dapat menggunakan IDE *online* seperti *Jupyter notebook* dan *google colab* (Alfarizi , 2023).dapat dilihat pada gambar 2.20.



Gambar 2. 20. *Python* (Alfarizi, 2023).

2.12. Langkah-langkah penggunaan *Python*

Python adalah bahasa pemrograman yang populer dan serbaguna yang digunakan di berbagai bidang, mulai dari pengembangan perangkat lunak hingga analisis data. Bab ini

akan menjelaskan langkah-langkah penggunaan *Python*, termasuk instalasi, penggunaan interpreter, pembuatan dan eksekusi skrip, serta sumber daya yang dapat digunakan untuk belajar *Python*. Berikut merupakan Langkah-langkah penggunaan:

1. Instalasi *Python*:
 - a. Unduh installer *Python* dari situs resmi *Python* (<https://www.Python.org/downloads/>)
Pilih versi yang sesuai dengan sistem operasi Anda.
 - b. Ikuti petunjuk instalasi yang muncul di layar.
2. Mengonfirmasi Instalasi
Buka terminal atau command prompt (cmd) dan ketik *Python --version* atau *Python3 version* untuk memastikan *Python* telah terinstal dengan benar.
3. Menggunakan *Python* Interpreter
 - a. Buka terminal atau *command prompt* dan ketik *Python* atau *Python3* untuk memulai *Python shell* interaktif.
 - b. Gunakan *shell* ini untuk mengeksekusi perintah *Python* satu per satu.
4. Membuat dan Menjalankan Skrip *Python*
 - a. Buat file *Python* dengan ekstensi *.py* menggunakan teks editor atau IDE.
 - b. Tulis kode *Python* di dalamnya.
 - c. Jalankan Skrip dengan perintah *Python nama_skrip.py* atau *Python3 nama_skrip.py*.
5. Menggunakan *Package* dan *Libraries*
 - a. Instal dan kelola paket-paket *Python* menggunakan pip, manajer paket *Python*.
 - b. Contoh: *pip install nama_paket* untuk menginstal sebuah paket.
6. Belajar *Python*.
 - a. Pelajari dasar-dasar *Python* seperti variabel, tipe data, perulangan, percabangan, dan

fungsi.

- b. Pelajari modul dan pustaka bawaan *Python*.
- c. Pahami konsep pemrograman berorientasi objek.
- 7. Praktik dan Proyek
 - a. Latihan dengan menulis kode *Python* untuk tugas dan tantangan pemrograman.
 - b. Mulailah dengan proyek-proyek kecil dan tingkatkan kompleksitasnya seiring berjalannya waktu.

8. Rujukan dan Sumber Belajar

Manfaatkan sumber-sumber belajar *online* seperti buku, tutorial, video, dan kursus *online* untuk meningkatkan pemahaman Anda tentang *Python*.

2.13. *Library Python*

Untuk memahami lebih lanjut tentang penggunaan *Library Python* dalam Dalam Deteksi Tingkat Kerusakan Jalan Dengan Pendekatan *deep Learning*, berikut beberapa contoh *Library Python* yang cukup populer yang dapat digunakan dalam penelitian ini:

- 1. *Numpy: Library* ini berisikan fungsi matematika dengan kompleksitas tinggi dapat digunakan untuk memproses array dan matriks multidimensi yang besar. *NumPy* juga sangat berguna untuk menangani aljabar linier, transformasi *Fourier*, dan bilangan acak
- 2. *Scikit Learn: Library* ini menyediakan banyak algoritma pembelajaran tanpa pengawasan dan pengawasan. Itu dibangun di atas beberapa teknologi yang mungkin sudah Anda kenal, seperti *NumPy*, *panda*, dan *Matplotlib*.
- 3. *Pandas: Library Python* paling populer yang digunakan untuk analisis data dengan dukungan untuk struktur data yang cepat, fleksibel, dan ekspresif yang dirancang

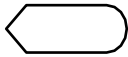
untuk bekerja pada data "relasional" atau "berlabel". *Pandas* hari ini adalah *library* yang tak terelakkan untuk menyelesaikan analisis data dunia nyata yang praktis dengan *Python*.

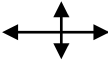
4. *Tensorflow*: *Library* yang digunakan untuk membangun dan menerapkan model pembelajaran mesin, terutama model pembelajaran mendalam.

2.14. *Flowchart*

Flowchart merupakan urutan logika dari suatu prosedur pemecahan masalah, sehingga *flowchart* merupakan langkah-langkah penyelesaian masalah yang dituliskan dalam simbol-simbol tertentu. Diagram alir ini akan menunjukkan alur didalam program secara logika. *Flowchart* bertujuan untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai rapi dan jelas menggunakan simbol-simbol standar (Charis Noijs, 2023). Berikut dapat dilihat pada tabel 2.3.

Tabel 2.3. Simbol Dan Fungsi *Flowchart* (Charis Noijs, 2023).

No.	Simbol	Fungsi
1.		Terminal, untuk memulai dan mengakhiri suatu proses / kegiatan.
2.		Proses, suatu yang menunjukan setiap pengolahan yang dilakukan oleh komputer.
3.		<i>Input</i> , untuk memasukan hasil dari suatu proses.
5.		<i>Display</i> , <i>output</i> yang ditampilkan dilayar terminal
6.		<i>Connector</i> , suatu prosedur akan masuk atau keluar melalui simbol ini dalam lembar yang sama.

7.		<i>Off Page Connector</i> , merupakan <i>symbol</i> masuk atau keluarnya suatu prosedur pada kertas lembar lain.
8.		Arus <i>Flow</i> , simbol ini digunakan untuk menggambarkan arus proses dari suatu kegiatan lain.
9.		<i>Hard Disk Storage</i> , <i>input output</i> yang menggunakan <i>hard disk</i> .
10.		<i>Predifined Process</i> , untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur.

BAB III

METODE PENELITIAN

3.1. Objek dan Waktu Penelitian

Objek yang diteliti pada penelitian ini yaitu jalan, citra gambar dari jalan yang akan menjadi objek untuk analisis. Hasil foto yang diambil menggunakan kamera digital dan Rentang waktu penelitian harus mencakup periode ketika data citra digital atau foto jalan yang akan dianalisis diperoleh. Periode waktu di mana analisis dilakukan menggunakan CNN Penentuan waktu penelitian juga bisa mencakup waktu pengumpulan citra digital, pengolahan data, dan analisis. Penelitian yang dilakukan pada semester ganjil tahun ajaran 2023/2024.

3.2. Sifat Jenis dan Pendekatan Penelitian

Jenis penelitian yang akan dilakukan menerapkan kuantitatif, kombinasi merupakan penelitian yang menggabungkan penggunaan pendekatan penelitian kualitatif dan kuantitatif dalam penelitian ilmiah. Pendekatan kualitatif penulis gunakan untuk menganalisis kajian *deep learning* dan penerapannya dalam proses pembelajaran.

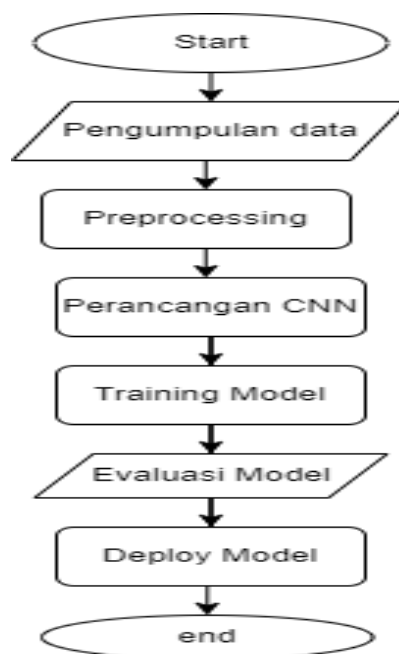
Penelitian ini memiliki sifat model sequensial, karena akan menggambarkan karakteristik data yang sudah dikumpulkan, mengacu pada pendekatan penelitian yang dilakukan secara berurutan atau bertahap. penelitian sequential dapat melibatkan langkah-langkah diambil secara berurutan, dengan setiap langkah bergantung pada hasil langkah sebelumnya.

Pendekatan kombinasi dalam klasifikasi kerusakan jalan yang rusak dapat memadukan keunggulan dari metode kualitatif dan kuantitatif untuk memberikan pemahaman yang lebih holistik dan mendalam. Dalam hal ini pendekatan metodologi

kombinasi yang dapat diterapkan dalam penelitian klasifikasi kerusakan jalan fase kualitatif awal untuk mendapatkan pemahaman mendalam tentang kerusakan jalan. Ini bisa melibatkan observasi lapangan, wawancara dengan ahli atau pengguna jalan, atau analisis dokumen terkait.

3.3. Diagram alir

Secara garis besar penelitian ini akan dilaksanakan seperti diagram alir dapat dilihat pada gambar 3.1.

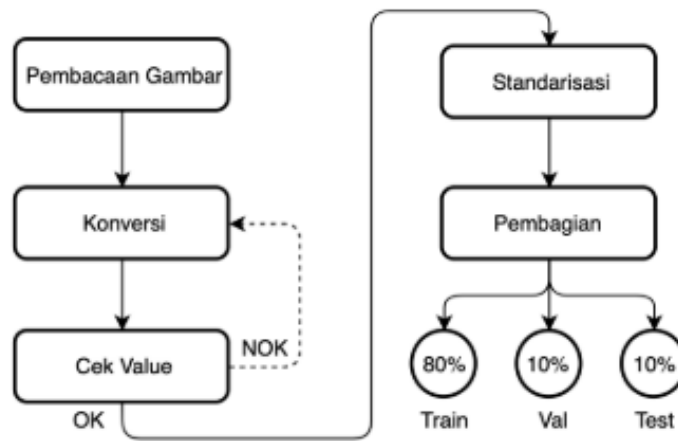


Gambar 3.1. diagram alur Penelitian.

1. Metode pengumpulan data

Data yang digunakan dalam penelitian ini berupa citra jalan yang mencakup berbagai kondisi jalan antara lain jenis lubang, jenis retak garis (*Longitudinal dan transversal*), serta jenis retak *non* garis, Data yang didapatkan yaitu merupakan data primer. Pengambilan data dilakukan saat kondisi langit cerah dan posisi lurus menghadap permukaan kerusakan jalan, menggunakan kamera *handphone* dengan resolusi yang cukup tinggi untuk memungkinkan

citra jalan yang akan di ambil dapat dideteksi dengan akurat. Mengumpulkan *dataset* yang mencakup berbagai tingkat kerusakan jalan dengan metadata spasial, seperti lokasi geografis, jenis sampel data yang ada dapat dilihat pada gambar 3.2.



Gambar 3. 2. proses pengolahan data.

Dalam proses persiapan data, Resolusi image dari dataset bervariasi sesuai dengan inputan kamera. menggunakan bahasa pemrograman *Python* untuk melakukan proses pengolahan data. pembacaan image dimulai dengan menyesuaikan jenis image dengan standar yang akan diaplikasikan. Pada penelitian ini standar image yang digunakan adalah format kompresi JPEG. Pada tahap standarisasi data, diasumsikan bahwa semua data sudah rapi (*without null*), bersih dan sudah cukup merepresentasikan kondisi sebaran data ideal. Pada proses standarisasi semua data akan diberikan label sesuai kelas yang akan diprediksi. Pada penelitian ini, data dibagi ke dalam tiga kondisi yaitu jalan dengan kondisi rusak ringan dan rusak berat.

2. *Pre-processing Data*

Preprocessing citra dengan mengubah ukuran citra gambar jalan rusak menjadi ukuran yang sama. Langkah selanjutnya dilakukan pembagian data yang telah dikumpulkan menjadi dua, yaitu data latih (*training*) dan data uji (*testing*). Langkah selanjutnya dilakukan

pelabelan pada masing-masing jalan rusak untuk pengenalan kerusakan jalan rusak ringan, dan berat pada data latih dan data uji.

a. Normalisasi

Proses mengubah skala data agar berada dalam rentang tertentu, biasanya 0 hingga 1 atau -1 hingga 1. Normalisasi dilakukan untuk memastikan bahwa setiap fitur berkontribusi secara proporsional dalam proses pelatihan model, sehingga mempercepat konvergensi dan meningkatkan stabilitas pelatihan model.

b. Augmentasi data

Teknik yang digunakan untuk memperbesar ukuran dan variasi dataset dengan melakukan transformasi pada data yang ada. Ini berguna dalam *deep learning*, terutama ketika data yang tersedia terbatas. Augmentasi data membantu meningkatkan generalisasi model dan mengurangi risiko *overfitting* dengan menciptakan versi baru dari data asli melalui berbagai transformasi.

3. Perancangan CNN

CNN adalah jaringan saraf tiruan yang mengekstrak fitur dari sebuah gambar input. Fitur yang diekstraksi akan digunakan oleh model untuk melakukan klasifikasi. Klasifikasi pada CNN kemudian bekerja berdasarkan ekstraksi fitur gambar jalan dan menghasilkan *output* sebuah prediksi. Jaringan CNN dalam melakukan ekstraksi fitur memiliki beberapa lapisan (*layer*) konvolusi dan kumpulan lapisan penyatuan (*pooling*) sebuah model yang digunakan untuk melatih data dalam mengenali objek yang diinginkan. Penelitian ini menggunakan jaringan VGG dalam melakukan ekstraksi fitur dan menjalankan training, Model VGG, atau VGGNet, yang menerapkan 16 *layers* NN juga disebut sebagai VGG16. VGG model merupakan jaringan saraf *Convolutional*. *Input image size* pada penelitian ini

adalah 150x150 *pixel* menyesuaikan dengan *input* VGG model. Secara aksitektur, dapat dijelaskan sebagai berikut:

a. *Input Layer*: Gambar input diubah ukurannya menjadi 150x150 *pixel* dan memiliki 3 kanal warna (merah, hijau, biru).

b. *Feature Learning*

Pada bagian ini lapisan yang dirancang untuk mengidentifikasi dan mengekstrak fitur-fitur penting dari data *input*, seperti *citra*. Berikut adalah bagaimana proses *feature learning* umumnya terjadi dalam CNN:

1. *Convolutional layer*

a. Filter/kernel

Berikut contoh ilustrasi proses konvolusi pada citradengan filter atau kernel, operasi *convolution* Citra dengan sebuah filter atau kernel seperti yang digambarkan pada Gambar 1 terdapat 1 *channel input* dengan ukuran matriks 4×4 dengan ukuran matriks dari filter yaitu 2×2 dan menghasilkan *output matriks* hasil operasi *convolution* dengan ukuran 3×3 . Operasi konvolusi dilakukan dengan menggeser filter diseluruh citra *input* dan menghitung titik dari filter dengan bagian citra yang tumpang tindih dapat dilihat pada gambar 3.3.

3	1	4	2
5	9	2	6
5	3	5	8
9	7	9	3

filter/kernel	
1	0
0	1

Gambar 3.3. *Convolution Matirx*.

Langkah selanjutnya dilakukan proses perhitungan disetiap *channel* dikalikan dengan kernel ukuran 2x2 pada gambar 3.2 tahapan ini dilakukan dengan berulang dengan

pergeseran kernel sebanyak *stirde*s di setiap channelnya sehingga di dapatkan perhitungan di setiap channel dengan nilai sebagai berikut dapat dilihat pada gambar 3.4.

3	1	4	2
5	9	2	6
5	3	5	8
9	7	9	3

$$(1) = (3*1) + (1*0) + (5*0) + (9*1) \\ = 3 + 0 + 0 + 9 = 12$$

Gambar 3.4. *Convolution* pada posisi ke1.

b. *Stirde*

Setelah itu di lakukan pergeseran pada Pergeseran piksel ini dilakukan secara *horizontal* dan vertikal. Sebagai contoh jika *stride* bernilai 1 maka akan bergeser sebanyak 1 piksel secara *horizontal* dan vertikal.dapat di lihat pada gambar 3.5.

3	1	4	2
5	9	2	6
5	3	5	8
9	7	9	3

$$(2) = 1*1 + 4*0 + 9*0 + 2*1 \\ = 1 + 0 + 0 + 2 = 3 \\ (3) = 4*1 + 2*0 + 2*0 + 6*1 \\ = 4 + 0 + 0 + 6 = 10$$

Gambar 3.5. *Convolution* pada posisi ke2.

c. *Padding*

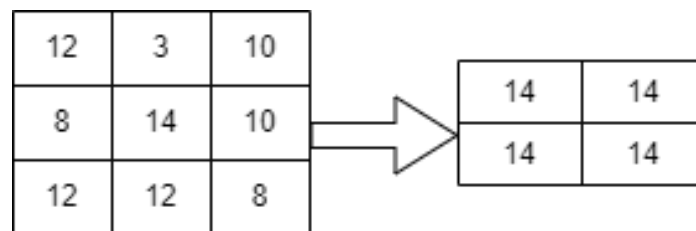
Setelah itu akan menghasilkan matrik dengan ukuran 3×3 menggunakan stride 1 tanpa menggunakan *padding* sehingga operasi konvolusi akan lebih kecil dari *input*. Ini terjadi karena konvolusi dilakukan dengan menggeser filter (kernel) di atas *input*, dan setiap pergeseran menghasilkan pengurangan ukuran dapat dilihat pada gambar 3.6.

12	3	10
8	14	10
12	12	8

Gambar 3.6. hasil dari *convolution*.

2. *Pooling Layer*

Setelah itu *pooling layer* untuk mempercepat proses komputasi tanpa harus kehilangan piksel-piksel yang memiliki informasi yang berguna. Menggunakan *max pooling* dengan *pooling 2×2* dengan *stride 1* untuk mencari nilai tertinggi dari piksel-piksel tersebut seperti yang digambarkan pada gambar 3.7.

Gambar 3.7. Hasil *pooling layer*.

3. *Rectified Linear Units (ReLU)*

Fungsi (ReLU) layer yang merupakan fungsi aktivasi yang digunakan dalam jaringan saraf tiruan, terutama dalam model *deep learning* seperti CNN. Fungsi ReLU membantu model untuk memperkenalkan non-linearitas ke dalam jaringan, yang memungkinkan model untuk belajar dan memodelkan hubungan yang kompleks dalam data. fungsi aktivasi $f(x) = \max(0, x)$ fungsi ReLU sangat sederhana, yaitu mengembalikan nilai *input* jika nilai tersebut positif, atau 0 jika nilai *input* negatif.

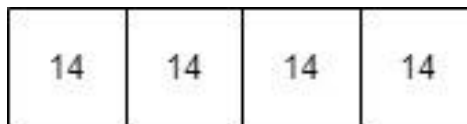
c. *Clasification*

Pada bagian ini dimana model digunakan untuk mengidentifikasi kategori atau label dari data *input*. Proses klasifikasi dalam CNN melibatkan beberapa tahap penting setelah

ekstraksi fitur melalui lapisan konvolusi dan *pooling*. Berikut adalah tahapan tentang bagaimana klasifikasi bekerja dalam CNN:

1. *Flaten*

Setelah proses konvolusi dalam arsitektur CNN, langkah berikutnya adalah mengubah data keluaran dari *layer* konvolusi dan *pooling* menjadi bentuk yang bisa digunakan oleh *fully connected layer* mengubah matriks multi-dimensi (*output* dari *layer* konvolusi/*pooling*) menjadi vektor satu dimensi. karena *layer fully connected* (*dense layer*) dalam *neural network* bekerja dengan vektor satu dimensi dapat dilihat pada gambar 3.8.



Gambar 3.8. Hasil konvolusi menjadi vektor 1D.

2. Lapisan *Fully Connected Layers*

Vektor yang telah diratakan diteruskan ke satu atau lebih lapisan *fully connected* untuk prediksi akhir. berikut perhitungan secara manual diproses oleh *fully connected layer* (*hidden layer*). *Fully connected layer* pertama (*hidden layer*) memiliki 3 *neuron*. Setiap *neuron* dalam *hidden layer* akan menerima *input* dari semua nilai dalam vektor yang di-flatten:

$$\text{Input Layer} = x_1 x_2$$

$$\text{Hiden Layer} = m_1 m_2 m_3$$

$$\text{Bobot: } w_1 = 0,7 \ w_2=0,5 \ w_3=0,3 \ w_4 = 0,6$$

Perhitungan Neuron 1 diHiden Layer

$$\delta 1 = W_1 \times X_1 + W_2 \times X_2 = 0,7 \times 14 + 0,5 \times 14 = 9,8 + 7 = 16.8$$

$$m1 \frac{1}{1 + e^{-16,8}} = 0,99$$

Perhitungan Neuron 2 dihidden Layer

$$\delta 2 = W_2 \times X_1 + W_3 \times X_2 = 0,5 \times 14 + 0,3 \times 14 = 7 + 4,2 = 11,2$$

$$m2 \frac{1}{1 + e^{-11,2}} = 0,99$$

Perhitungan Neuron 3 dihidden Layer

$$\delta 3 = W_3 \times X_1 + W_4 \times X_2 = 0,3 \times 14 + 0,6 \times 14 = 4,2 + 8,4 = 12,6$$

$$m3 \frac{1}{1 + e^{-12,6}} = 0,99$$

Dan perhitungan nilai output pada lapisan output adalah sebagai berikut.

$$\delta 1 = W_5 \times m_1 + W_6 \times m_2 + w_7 \times m_3 = 0,3 \times 0,99 + 0,5 \times 0,99 + 0,9 \times 0,99$$

$$\delta 2 = 0,29 + 0,49 + 0,89 = 1,69$$

3. Los layer

Setelah itu bagian akhir dari jaringan setelah lapisan *fully connected* atau lapisan *output*. Hasil dari lapisan *output* (seperti probabilitas dari *softmax*) dibandingkan dengan label yang sebenarnya menggunakan fungsi *loss*. Hasil perhitungan *loss* ini kemudian digunakan untuk memperbarui bobot-bobot di seluruh jaringan.

Aktivasi menggunakan fungsi sigmoid

$$x_0 \frac{1}{1 + e^{-1,69}} = 0,84$$

4. Pelatihan Model

Setelah dilakukan perancangan model CNN, tahapan selanjutnya adalah melakukan pelatihan model menggunakan data pelatihan dengan fitmodel dimana metode yang

di gunakan untuk melakukan fitting pada Model dengan menggunakan data latih yang telah dikumpulkan sebelumnya. Selanjutnya, pada proses pengujian model digunakan jumlah *epoch* (iterasi) untuk menentukan berapa kali jaringan akan melakukan pelatihan. Pada tahap ini terdapat fungsi *loss* yang digunakan untuk melihat performa dari model CNN.

5. *Loss*

metrik yang digunakan untuk mengukur seberapa baik atau buruk model prediksi Anda selama pelatihan. Dalam konteks jaringan saraf seperti CNN, *loss* memberikan indikasi tentang seberapa jauh prediksi model dari label yang benar. Semakin kecil nilai *loss*, semakin baik kinerja model pada data yang diberikan, fungsi *loss* ini kemudian digunakan untuk mengupdate bobot model dalam proses training melalui *backpropagation*.

6. Analisis Akurasi

Analisis akurasi pada evaluasi model adalah proses yang sistematis dan terstruktur untuk memastikan bahwa model yang telah dilatih bekerja dengan baik pada data baru yang belum pernah dilihat sebelumnya. Dengan mengikuti tahapan ini, kita dapat memperoleh gambaran yang jelas tentang kinerja model dan langkah-langkah yang perlu diambil untuk memperbaikinya.

7. Evaluasi Model

Proses yang penting dalam pengembangan dan penilaian model machine learning atau statistik untuk memahami seberapa baik model tersebut berperforma dalam memecahkan masalah yang dihadapi. Evaluasi model melibatkan penggunaan berbagai metrik dan teknik untuk mengukur kinerja model dan membuat penilaian

yang tepat tentang kemampuannya untuk membuat prediksi yang akurat pada data yang belum pernah dilihat sebelumnya.

8. *Deployment*

Setelah model CNN dievaluasi dan menunjukkan performa yang memadai, langkah selanjutnya adalah melakukan *deployment*. *Deployment* adalah proses menempatkan model yang telah dilatih ke dalam lingkungan produksi sehingga bisa digunakan untuk membuat prediksi pada data baru memastikan model terintegrasi dengan baik dengan sistem lain yang memerlukan prediksi, alat dan bahan penelitian.

Dalam melakukan penelitian ini, ada beberapa spesifikasi alat penelitian yang harus dipenuhi. Spesifikasi alat penelitian maksudnya adalah standar minimal dari alat (*tools*) yang digunakan sebagai wadah utama dalam memanfaatkan teknologi *deep learning* dalam deteksi tingkat kerusakan jalan. Alat yang digunakan antara lain sebagai berikut:

3.4.1. Perangkat keras (*hardware*)

Adapun spesifikasi perangkat keras (*hardware*) yang digunakan pada penelitian ini sebagai penunjang proses perancangan dan pembuatan sistem selama penelitian berlangsung. Spesifikasi *hardware* dapat dilihat pada tabel 3.1.

Tabel 3.1 spesifikasi *Hardware*

No	Jenis	Spesifikasi
1.	Leptop	HP Spectre x360 13
2.	<i>Processor</i>	Intel Core i5 gen 8
3.	Memori (RAM)	4 GB
4.	SSD	512 GB
5.	<i>System Type</i>	64-bit

3.4.2. Perangkat Lunak (*software*)

Selain kebutuahn *hardware* yang memumpuni penulis juga membutuhkan *software* yang mendukung si penulis untuk melakukan sebuah perancangan dan pembuatana sistem,

untuk deteksi kerusakan jalan. Spesifikasi *software* dapat dilihat pada tabel 3.2.

Tabel 3. 2. Spesifikasi *Software*

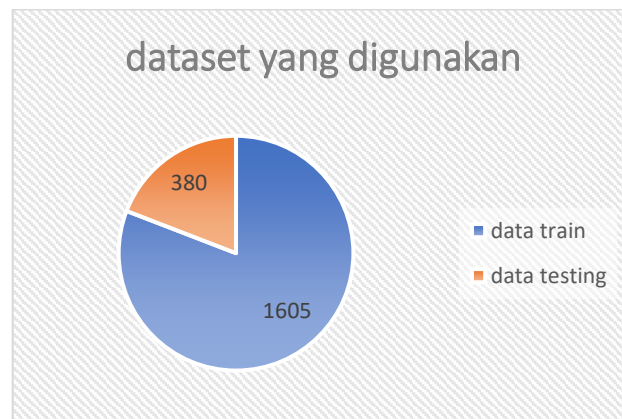
No	Jenis	Spesifikasi	Keterangan
1.	Sistem Operassi	<i>Windows 11</i>	Sistem Operasi yng digunakan selama penelitian
2.	<i>Bahasa Pemograman</i>	<i>Python</i>	Digunakan untuk pembuatan System
3.	<i>Teks editor</i>	<i>Visual Studio Code</i>	menulis dan menjalankan <i>script Python</i>
4	<i>Library Pandas</i>	<i>import pandas as pd:</i>	untuk analisis data yang menyediakan struktur data fleksibel bernama DataFrame
5.	<i>Library scikit-learn</i>	<i>train_test_split</i>	mengimplementasikan algoritma CNN Membagi dataset menjadi set pelatihan dan set pengujian
6.	<i>Library Numpy</i>	<i>Import Numpy as pd</i>	untuk mengelola data dalam bentuk matriks
7.	<i>Library tensorflow</i>	<i>Import tensorflow as tf</i>	uk membangun dan menerapkan model pembelajaran mesin, terutama model pembelajaran mendalam

BAB IV

HASIL DAN PEMBAHASAN

4.1. Analisis Data

Data yang di gunakan pada penelitian ini menggunakan dataset terbuka (*open dataset*) yang di ambil dari *Platform Roboflow* yaitu dataset *Road Damage Indonesia*. *Dataset* tersebut terdiri dari 2 kelas yang meliputi rusak ringan dan rusak berat dengan jumlah data *Train* 1605, data *Test* 380, dapat di lihat pada gambar 4.1.



Gambar 4.1 Dataset yang digunakan.

4.2. Pre-prosesing

Pada tahap *pre-processing* data yang di kumpulkan dari *road damage indonesia* lalu diberikan Label pada tiap gambar sesuai dengan kategori kerusakan dimana terdapat dua kategori yaitu rusak ringan dan rusak berat, terdapat langkah-langkah yang dilakukan untuk mempersiapkan data mentah sebelum digunakan untuk melatih model *machine learning* berikut tahapannya:

1. Normalisasi

Pada tahapan ini mengubah nilai pixel gambar dari rentang $[0,255]$ menjadi $[0,1]$, ini Dimana dilakukan dengan membagi setiap nilai piksel dengan 255. Dimana membantu

jaringan saraf untuk beroperasi dalam mempercepat konvergensi model selama pelatihan

2. Augmentasi Data

Pada augmentasi data dilakukan untuk meningkatkan ukuran dan keragaman dataset dengan membuat variasi buatan pada gambar asli. dimana membantu model menjadi lebih *robust* dan mengurangi *overfitting*.

a. *Rotation* (Rotation Range =40)

Pada tahapan ini dilakukan pengizinan rotasi gambar secara acak hingga 40 derajat, sehingga Model dapat mengenali Objek Yang diputar.

b. *Shift* (*Shear_Range*=0.2)

Pada tahapan ini dilakukan pengizinan pergeseran gambar secara *horizontal* dan *vertikal* hingga 20% dari ukuran gambar. Ini membantu model mengatasi variasi posisi objek dalam gambar.

c. *Shear* (*Shear_Range*=0.2)

Pada tahapan ini dilakukan penerapan shear transformation, yang dapat mengubah bentuk gambar secara miring. untuk membantu model mengenali objek yang telah mengalami deformasi.

d. *Zoom Range*(*zoom_range*=0.2)

Pada tahapan ini dilakukan pengizinan *zoom in zoom out* pada gambar hingga 20%, membantu model mengenali objek dengan berbagai skala.

e. *Horizontal*

Pada tahapan ini dilakukan pembalikan gambar secara *horizontal*, berguna untuk augmentasi data dan membuat model mengenali objek meskipun orientasinya dibalik.

f. *Nearest*

Pada tahapan ini dilakukan, pengisian area kosong yang mungkin muncul akibat augmentasi dengan *pixel* terdekat.

Berdasarkan *preprocessing* data diatas, berikut kodingan implementasi *preprocessing* data pada dataset, dapat dilihat pada gambar 4.2.

```

train_datagen = ImageDataGenerator(rescale=1./255,
                                   rotation_range=40,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True,
                                   fill_mode='nearest')

test_datagen = ImageDataGenerator(rescale=1./255)

```

Gambar 4.2. *Preprocessing* data.

Berdasarkan gambar di atas menjelaskan, dimana *preprocessing* data diatur dengan *rescale=1./255*, *rotation_range=40*, *width_shift_range=0.2*, *height_shift_range=0.2*, *shear_range=0.2*, *zoom_range=0.2*, *horizontal_flip=True*, *fill_mode='nearest'*. Tahapan ini dibuat agar data yang digunakan siap diolah oleh algoritma CNN. Berikut gambar *preprocessing* pada *width_Shift_range* dapat dilihat pada gambar 4.3.

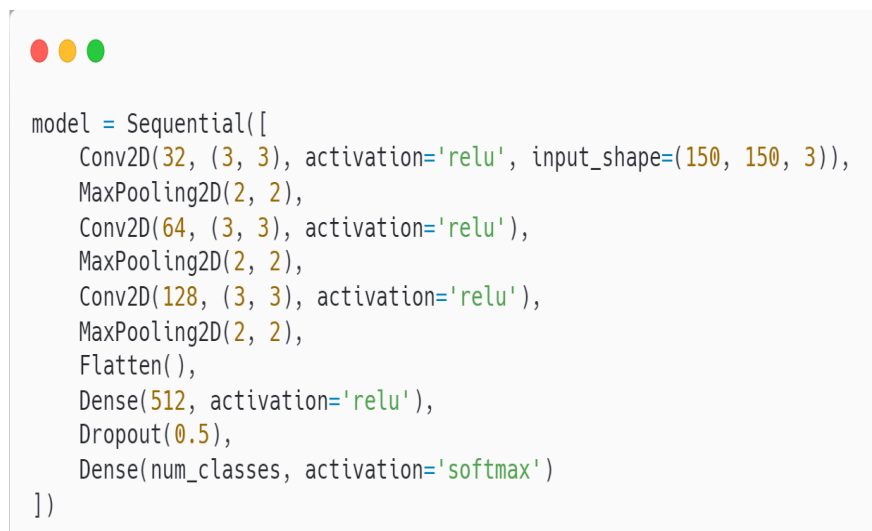


Gambar 4.3. *Preprocessing* widthshift Range.

Pada Gambar 4.3. merupakan parameter untuk menentukan rentang pengujian *horizontal* dalam pixel Saat menggunakan augmentasi gambar, setiap gambar dalam set data pelatihan dapat diubah secara *horizontal* dalam jarak tertentu, baik ke arah kanan maupun kiri. Rentang ini menentukan seberapa jauh gambar dapat digeser ke kiri atau ke kanan.

4.3. Implementasi CNN

Pada Tahapan ini pembuatan model akan terdiri dari proses konvolusi yang disertai fungsi aktivasi dan proses *pooling*. Banyaknya proses ini disesuaikan dengan kebutuhan peneliti. Model dibuat dengan memuat beberapa jenis lapisan yang berbeda, antara lain lapisan konvolusi (*layer_conv_2d*), lapisan *pooling*, lapisan *dropout*, lapisan *flatten* dan lapisan *dense*. Proses konvolusi ini dilakukan sebanyak 4 kali ditunjukkan oleh banyaknya *convolution layer* yang digunakan. Pada umumnya 2 sampai 3 lapisan cukup untuk mendapatkan model klasifikasi yang memiliki akurasi tinggi dapat dilihat pada gambar 4.4.



```

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    MaxPooling2D(2, 2),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(num_classes, activation='softmax')
])

```

Gambar 4.4. Implementasi Model CNN.

pada penelitian ini ada beberapa lapisan dalam melatih model dan mengetahui kinerja model tersebut. Fungsi aktivasi yang kami gunakan pada penelitian ini adalah ReLu

(*Rectified Linier Unit*) yang digunakan tahap pelatihan menjadi lebih cepat. Ukuran kernel yang digunakan untuk setiap lapisan konvolusi sebesar 3×3 . Pada proses *pooling* digunakan metode *maxpooling* dengan ukuran filter 2×2 sehingga nilai maksimum pada area piksel tersebut yang akan dipilih pada setiap pergeseran.

Jumlah filter yang digunakan dalam *convolution layer* yang pertama dan kedua adalah 32. Filter yang digunakan pada *convolution layer* ketiga dan keempat adalah 64. Penggunaan jumlah filter yang lebih banyak pada dua lapisan konvolusi terakhir disebabkan oleh ukuran *input* pada kedua lapisan tersebut lebih kecil sehingga dibutuhkan lebih banyak filter untuk mengesktrak informasi citra.

4.4. Pelatihan Model

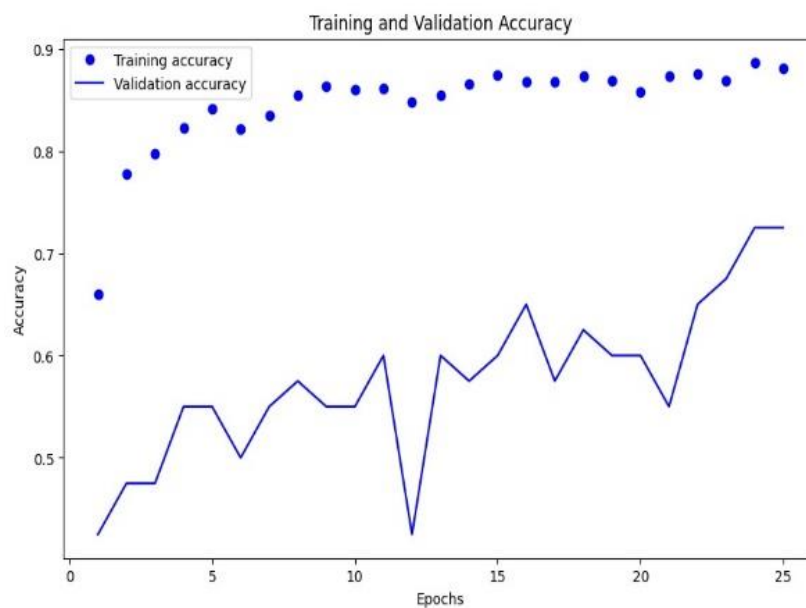
Setelah model dibangun, tahapan selanjutnya kita melatih dengan data kerusakan jalan yang telah di proses setelah itu *dataset* di bagi menjadi *training set*, *validation set*, dan *test set*. Ke dalam model dengan fit model. Dengan menggunakan *epoch* 25, *batch_size* = 100, digunakan untuk menentukan berapa kali jaringan akan melihat seluruh kumpulan data setelah pelatihan selesai evaluasi kinerja model untuk menilai kinerjanya pada data yang belum dilihat sebelumnya lalu visualiasasi dan analisis hasil pelatihan untuk mengetahui bagaimana model belajar selama pelatihan, dapat di lihat pada tabel 4.1.

Tabel 4.1 *Acurarcy* dan *Loss*

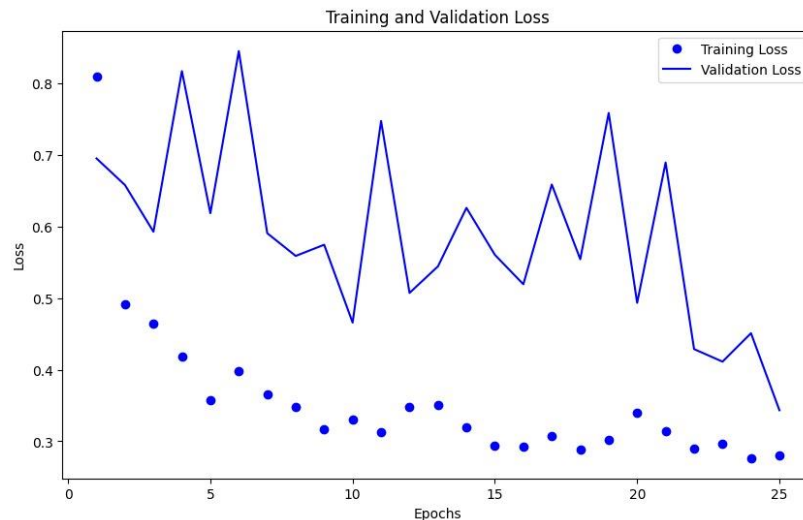
epoch	accuracy	Val Acuracy	Los	Val Loss
1/25	0.6604	0.4250	0.8100	0.6951
2/25	0,7775	0.4749	0.4915	0.6579
3/25	0,7975	0.4749	0.4637	0.5927
4/25	0,8224	0.5500	0.4185	0.8171
5/25	0.8417	0.5500	0.3574	0.6186
6/25	0.8218	0.5500	0.3974	0.8451
7/25	0.8348	0.5749	0.3653	0.5906
8/25	0.8542	0.5500	0.3474	0.5589

9/25	0.8635	0.5500	0.3173	0.5745
10/25	0.8598	0.6000	0.3300	0.4658
11/25	0.8610	0.4250	0.3127	0.7474
12/25	0.8473	0.6000	0.3485	0.5072
13/25	0.8542	0.5749	0.3510	0.5444
14/25	0.8654	0.6000	0.3192	0.6261
15/25	0.8741	0.6499	0.2943	0.5604
16/25	0.8672	0.5749	0.2926	0.5193
17/25	0.8679	0.6000	0.3074	0.6587
18/25	0.8728	0.6000	0.2877	0.5543
19/25	0.8685	0.5500	0.3015	0.7587
20/25	0.8573	0.6499	0.3396	0.4934
21/25	0.8735	0.6750	0.3137	0.6894
22/25	0.8753	0.7250	0.2903	0.4288
23/25	0.8691	0.7250	0.2970	0.4112
24/25	0.8866		0.2766	0.4510
25/25	0.8803		0.2799	0.3433

Pada Tabel tersebut Nilai *batch size* menunjukkan jumlah pelatihan dalam satu *forward/backward pass*. Semakin tinggi nilai *batch size* maka akan semakin banyak memori yang dibutuhkan. Hasil *loss* dan *accuracy* model yang terbentuk dapat dilihat pada Gambar 4.5. dan 4.6.



Gambar 4.5. Grafik Hasil Iterasi Nilai Akurasi.



Gambar 4.6. Grafik Hasil Iterasi Nilai Los.

Pada Gambar 4.5 dan 4.6 dapat dilihat bahwa pada iterasi ke 25 nilai akurasi yang dihasilkan pada data latih yaitu 0.880 dan nilai loss yang dihasilkan pada data latih yaitu 0,279, Berdasarkan hasil pengujian diperoleh informasi bahwa model yang dibuat mampu mengklasifikasikan jenis keretakan pada jalan dengan baik. Pergerakan loss yang mendekati nilai nol atau kurang dari satu dan akurasi yang terus meningkat menunjukkan hasil yang baik seiring dengan berjalannya *epoch*. Kondisi berhentinya proses pelatihan ditentukan dari banyaknya *epoch* yang digunakan, dalam hal ini yaitu sebanyak 25 *epoch*. Proses pembelajaran tersebut akan berhenti jika sudah memenuhi kondisi tersebut.

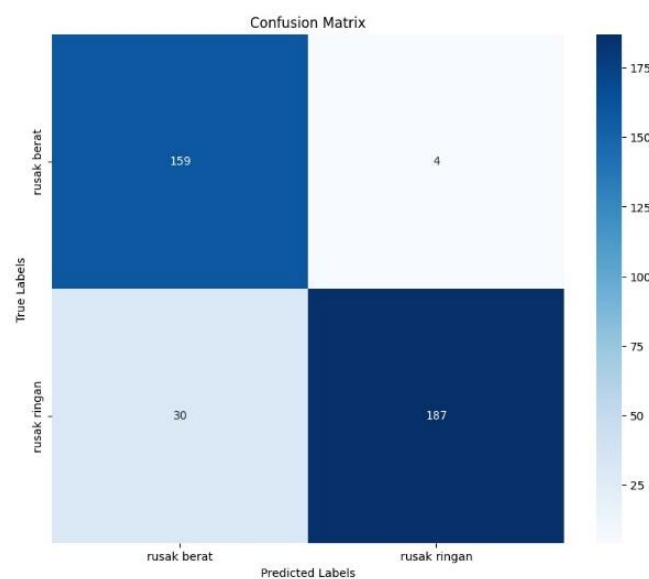
4.5. Evaluasi Model

Pada tahapan ini melakukan masuk pada Proses implementasi algoritma CNN menggunakan data uji untuk mengevaluasi model, berikut tahapan evaluasi model:

1. Akurasi (*Accuracy*)

Berdasarkan data hasil pelatihan model CNN pada setiap *epoch*. Data ini mencakup akurasi (*accuracy*) dan loss (*loss*) untuk kedua *dataset*, yaitu dataset pelatihan (*train*) dan dataset validasi (*val*) Dari *epoch* 1 hingga *epoch* 25, terlihat adanya peningkatan secara

Signifikan dari akurasi pada kedua dataset, baik dataset pelatihan maupun *dataset* validasi. Pada akhir pelatihan (*epoch* 25), akurasi pada dataset pelatihan mencapai 88.03%, sedangkan pada dataset validasi mencapai 72.50%. Akurasi pada dataset pelatihan (*train*) cenderung lebih tinggi dibandingkan dengan akurasi pada *dataset* validasi (*val*). Ini bisa menjadi indikasi adanya *overfitting* jika perbedaannya terlalu besar confusion matrix dapat dilihat pada gambar 4.7.



Gambar 4.7. Tabel *confusion matrix*.

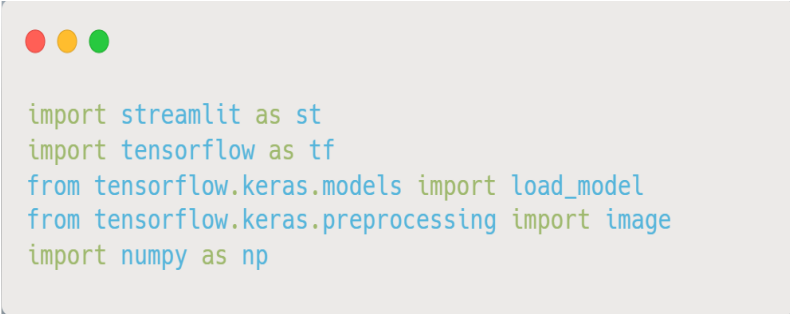
2. Loss

Dari *epoch* 1 hingga *epoch* 25, terlihat adanya penurunan secara umum dari nilai *loss* pada kedua dataset. Pada akhir pelatihan (*epoch* 25), nilai *loss* pada dataset pelatihan adalah 0.2799, sedangkan pada *dataset* validasi adalah 0.3433. *Loss* pada dataset pelatihan (*train*) cenderung lebih rendah dibandingkan dengan *loss* pada dataset validasi (*val*). Ini juga bisa menjadi indikasi adanya *overfitting* jika perbedaannya terlalu besar.

4.6. Deploy Model

Pada tahapan *deploy* model, model yang telah dibuat pada tahapan sebelumnya

akan dilihat dari nilai parameternya dari model yang telah dibuat. Ini melibatkan menyajikan model yang telah dilatih dan mengonversinya menjadi bentuk yang dapat digunakan untuk memastikan bahwa kinerjanya tetap optimal dan tidak ada degradasi performa yang signifikan. Pada tahapan ini masuk pada proses implementasi algoritma CNN untuk mengevaluasi model. Berikut tahapan implementasi evaluasi model pada gambar 4.8.



```
import streamlit as st
import tensorflow as tf
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
```

Gambar 4.8. Implementasi *Library Python*.

Pada Gambar 4.8 Merupakan Kode Sumber penggunaan *library* membuat *deploy* model Pada *Streamlit* dimana setiap bagian dari kode *Streamlit* yang memanfaatkan model Keras/*TensorFlow* untuk mengklasifikasikan gambar jalan rusak Setelah itu memuat model yang telah di latih. Dapat di lihat pada Gambar 4.9.



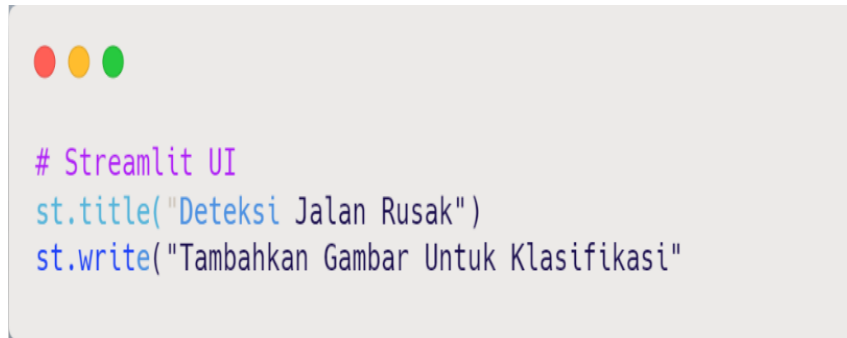
```
# Load the trained model
model = load_model('model_cnn.h5')

# Function to load and preprocess image
def load_and_preprocess_image(img_path):
    img = image.load_img(img_path, target_size=(150, 150))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array /= 255.0 # Rescale the image
    return img_array
```

Gambar 4.9. Implementasi Model CNN.

Pada Gambar 4.9 merupakan kode sumber Memuat Model yang telah di simpan pada

Direktori agar dapat di gunakan pada data Baru Setelah itu di buat antarmuka Pengguna *Streamlit* Dapat di Lihat Pada Gambar 4.10.



```
# Streamlit UI
st.title("Deteksi Jalan Rusak")
st.write("Tambahkan Gambar Untuk Klasifikasi")
```

Gambar 4.10. Antarmuka *Streamlit*.

Pada Gambar 4.10. Merupakan kode sumber Untuk Membuat Tampilan Interaksi pada *Streamlit* Dimana Menampilkan Judul Sistem yang di buat dan memberikan Instruksi Kepada Pengguna dapat dilihat pada gambar 4.11.



```
uploaded_file = st.file_uploader("Pilih Gambar...",
                                  type=["jpg", "jpeg", "png"])

if uploaded_file is not None:
    # Save the uploaded file temporarily
    with open("temp_image.jpg", "wb") as f:
        f.write(uploaded_file.getbuffer())

    # Display the uploaded image
    st.image(uploaded_file, caption='Gambar Berhasil
    Diunggah', use_column_width=True)
```

Gambar 4.11. Memuat file *Uploader*.

Pada Gambar 4.11. merupakan Kode Sumber Untuk Mengunggah file Gambar dengan Format tertentu dan Gambar tersebut akan di simpan sementara lalu akan menampilkan Gambar yang di unggah dengan keterangan “Gambar Berhasil Di unggah” Setelah itu dilanjutkan dengan penambahan atribut seperti *title*, *text_input* untuk Deteksi Jalan rusak, dan juga *button* sebagai pengekseskusi model yang akan diklasifikasi seperti Pada Gambar 4.12.

```

# Button to start the classification
if st.button("Deteksi"):
    # Load and preprocess the image
    img_array =
load_and_preprocess_image("temp_image.jpg")

    # Make a prediction
    prediction = model.predict(img_array)
    predicted_class = np.argmax(prediction,
axis=1)

    # Define class labels
    labels = ['Rusak Berat', 'Rusak Ringan']

    # Display the prediction result
    st.write("Hasil Klasifikasi: ")
    st.success(f"{labels[predicted_class[0]]}")

```

Gambar 4.12. Membuat *Input Button*.

Pada Gambar 4.12 membuat sebuah tombol di antarmuka *Streamlit* dengan label "Deteksi". Kemudian di proses menjadi *array Numpy* yang sesuai dengan model setelah itu model yang dilatih untuk membuat prediksi berdasarkan gambar yang di proses. setelah itu dilakukan pembacaan model yang kemudian dilanjutkan dengan penambahan atribut seperti *title*, *text_input* untuk Deteksi Jalan rusak, dan juga *button* sebagai pengeksekusi model yang akan diklasifikasi. Berikut tampilan dari *streamlit*. Dapat dilihat pada gambar 4.13.

Deteksi Jalan Rusak

Tambahkan Gambar Untuk Klasifikasi

Pilih Gambar...



Drag and drop file here
 Limit 20MB per file • JPG, JPEG, PNG

Browse files



subaim 1.jpg 158.5KB



Berdasarkan dengan SOP kerusakan jalan:

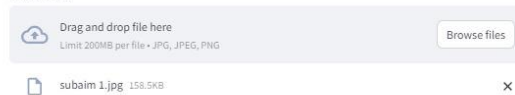
Gambar 4.13. Tampilan *title*, *text input*.

Pada Gambar 4.13 terdapat kolom yang di gunakan untuk menginputkan gambar yang akan di deteksi setelah itu gambar tersebut akan tersimpan untuk tampilan dapat di lihat pada gambar 4.14.

Deteksi Jalan Rusak

Tambahkan Gambar Untuk Klasifikasi

Pilih Gambar...



Berdasarkan dengan SOP kerusakan jalan:



Gambar Berhasil Diunggah

Gambar 4.14. Tampilan memuat Gambar.

Pada Gambar 4.14 terdapat *button* yang dimana pada saat mengklik *button* Deteksi maka model yang tadi di buat akan melakukan proses klasifikasi, setelah di lakukan proses klasifikasi maka akan menampilkan hasil deteksi dengan label yang sesuai di dalam sebuah kotak hijau seperti pada Gambar 4.15.

Kerusakan Ringan :

1. Retakan Halus <3mm
2. Alligator Cracking
3. Pengelupasan Permukaan Aspal/Beton yang Tidak Dalam
4. Lubang Kecil <13cm diameter
5. Penambalan Retakan

Kerusakan Berat :

1. Retakan Lebar >3mm
2. Pengelupasan atau Kerusakan dalam pada lapisan struktur
3. Lubang Besar >15cm
4. Rekonstruksi Lapisan Jalan

Deteksi

Hasil Deteksi Jalan:

Rusak Berat

Gambar 4.15. Tampilan Hasil deteksi kerusakan jalan.

BAB V

PENUTUP

5.1. Kesimpulan

Pada penelitian ini deteksi kerusakan jalan dengan metode CNN telah berhasil dilakukan dengan tingkat akurasi sebesar 88%. Model CNN yang digunakan terdiri dari empat lapisan konvolusi dengan ukuran filter 3x3, fungsi aktivasi yang digunakan yaitu *reLu*, dan 2 *pooling layer* dengan ukuran 2x2. Citra yang digunakan untuk penelitian ini yaitu sebanyak 1605 untuk data train dan data test 380. Berdasarkan hasil pelatihan diperoleh tingkat. Berdasarkan hasil pelatihan yang diperoleh tingkat akurasi dari model CNN yaitu sebesar 0,880 dengan *loss* 0,279. CNN terbukti sangat efektif dalam mengenali pola *visual* kompleks dalam gambar, yang membuatnya cocok untuk aplikasi seperti deteksi kerusakan jalan. Dengan kemampuan ini, CNN dapat mempelajari fitur-fitur penting yang membedakan kerusakan jalan dari kondisi jalan yang baik.

Penerapan model CNN untuk deteksi kerusakan jalan dalam menggunakan Streamlit menunjukkan sejumlah keunggulan dan langkah-langkah penting yang dapat diambil untuk mencapai hasil yang efektif. CNN sangat efektif dalam memproses dan menganalisis gambar karena arsitekturnya yang secara khusus dirancang untuk mendeteksi pola dan fitur *visual* seperti tepi, tekstur, dan bentuk.

5.2. Saran

Dalam penelitian mengenai deteksi tingkat kerusakan jalan menggunakan CNN, beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut atau peningkatan penelitian ini yaitu

1. Peningkatan *dataset* perluasan dan peningkatan kualitas *dataset* dengan mencakup

berbagai kondisi cuaca, pencahayaan, dan berbagai jenis kerusakan jalan. Penggunaan data augmentasi juga dapat membantu dalam memperbesar ukuran dataset secara efektif.

2. Dalam penelitian selanjutnya untuk mendapat akurasi yang tinggi dan menurunkan nilai *loss* dengan menambahkan lapisan konvolusi dan menambahkan fungsi optimasi selain itu pada penelitian lebih lanjut dapat digunakan dengan metode *Deep learning* lainnya sehingga diperoleh akurasi yang lebih baik.

DAFTAR PUSTAKA

- Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. 2023. Penggunaan *Python* Sebagai Bahasa Pemrograman untuk *Machine Learning* dan *Deep Learning*. Karya Ilmiah Mahasiswa Bertauhid (KARIMAH TAUHID), E-ISSN:2963-590X. Vol. 2 Issue. 1 Agustus, 2023
- Aptarila, G., Lubis, F., & Saleh, A. 2020. Analisis Kerusakan Jalan Metode SDI Taluk Kuantan - Batas Provinsi Sumatera Barat. *Siklus : Jurnal Teknik Sipil*, E-ISSN:2549-3973. Vol.6 No.2. Issue. 2, Oktober 2020
- Hadianto, N., Novitasari, H. B., & Rahmawati, A. 2019. Klasifikasi Peminjaman Nasabah Bank Menggunakan Metode *Neural Network*. *Jurnal Pilar Nusa Mandiri*, E-ISSN 1978-1946. Vol. 15 Issue. 2, September 2019
- Hu, L., & Ge, Q. 2020. *Automatic facial expression recognition based on MobileNetV2 in Real-time*. *Journal of Physics: Conference Series*, E-ISSN 1742-6596. Vol. 1549. Issue. 2 November 2020
- Lalu, K., Polri, L., & Perhubungan, K. 2020. Bab 1 1. pendahuluan 1.1. 1–5.
- Nugroho, P. A., Fenriana, I., & Arijanto, R. 2020. Implementasi *Deep Learning* Menggunakan *Convolutional Neural Network* (CNN) Pada Ekspresi Manusia. *Algor*, E-ISSN:2715-0577. Vol. 2, Issue. 1, September 2020
- Pramestya, R. H. 2018. Deteksi dan Klasifikasi Kerusakan Jalan Aspal Menggunakan Metode YOLO Berbasis Citra Digital. Institut Teknolgi Sepuluh November, E-ISSN
- Raup, A., Ridwan, W., Khoeriyah, Y., & Zaqiah, Q. Y. 2022. *Deep Learning* dan Penerapannya dalam Pembelajaran. E-ISSN:2614-8854. Vol. 5, No. 9 Issue 9, September 2022
- Rinaldi, N., Lestari, F., & Pramita, G. 2022. Identifikasi Kerusakan Jalan dan Alternatif Perbaikan Jalan Pada Ruas Jalan Tegineneng – Gunung Sugih Lampung. *Teknika Sains: Jurnal Ilmu Teknik*, E-ISSN: 2548-6403. Vol. 7, No. 01, Issue. 1, November 2022
- Sasmito, B., Setiadji, B. H., & Isnanto, R. 2023. Deteksi Kerusakan Jalan Menggunakan Pengolahan Citra *Deep Learning* di Kota Semarang. *Teknik*, E-ISSN:0852-1697. Vol. 44, Issue. 1, September 2023

- Yoga Triardhana, Bandi Sasmito, F. H. 2020. Identifikasi Kerusakan Jalan Menggunakan Metode *Deep Learning* (DI) Model *Convolutional Neural Networks* (CNN). *Jurnal Geodesi Undip*, ISSN:2337-845X. Vol. 10, No. 03, Agustus 2020
- Yunardhi, H. 2018. Analisa Kerusakan Jalan Dengan Metode Pci Dan Alternatif Penyelesaiannya (Studi Kasus : Ruas Jalan D.I. Panjaitan). *Jurnal Teknologi Sipil*, Vol. 2, No.2, *issue*. 2, November 2018
- Tempola, F., Wardoyo, R., Musdholifah, A., Rosihan, & Sumaryanti, L. (2024). *Classification of clove types using convolution neural network algorithm with optimizing hyperparamters. Bulletin of Electrical Engineering and Informatics*, E-ISSN 2302-9285. Vol.13 no 1,*issue*.1,February 2024