TESIS

ANALISIS KOMPARASI UNTUK DIAGNOSA PNEUMONIA BERDASARKAN HASIL CITRA CHEST X-RAY MENGGUNAKAN MODEL RESNET-50 DAN MOBILENETV2

Disusun oleh:

Nama : **Anggi Muhammad Rifa'i**
NIM : **21.51.2099**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2023

TESIS

ANALISIS KOMPARASI UNTUK DIAGNOSA PNEUMONIA BERDASARKAN HASIL CITRA CHEST X-RAY MENGGUNAKAN MODEL RESNET-50 DAN MOBILENETV2

COMPARATIVE ANALYSIS FOR DIAGNOSIS OF PNEUMONIA SYMTOMS USING CHEST X-RAY BASED ON RESNET-50 AND MOBILENETV2 MODELS

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama	**: Anggi Muhammad Rifa'i**
NIM	**: 21.51.2099**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2023

HALAMAN PENGESAHAN

ANALISIS KOMPARASI UNTUK DIAGNOSA PNEUMONIA BERDASARKAN HASIL CITRA CHEST X-RAY MENGGUNAKAN MODEL RESNET-50 DAN MOBILENETV2

COMPARATIVE ANALYSIS FOR DIAGNOSIS OF PNEUMONIA SYMTOMS USING CHEST X-RAY BASED ON RESNET-50 AND MOBILENETV2 MODELS

Dipersiapkan dan Disusun oleh

Anggi Muhammad Rifa'i

21.51.2099

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Senin, 3 Juli 2023

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 3 Juli 2023
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

ANALISIS KOMPARASI UNTUK DIAGNOSA PNEUMONIA BERDASARKAN HASIL CITRA CHEST X-RAY MENGGUNAKAN MODEL RESNET-50 DAN MOBILENETV2

COMPARATIVE ANALYSIS FOR DIAGNOSIS OF PNEUMONIA SYMTOMS USING CHEST X-RAY BASED ON RESNET-50 AND MOBILENETV2 MODELS

Dipersiapkan dan Disusun oleh

Anggi Muhammad Rifa'i

21.51.2099

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Senin, 3 Juli 2023

Pembimbing Utama

Prof. Dr. Emo Utami, S.Si., M.Kom.
NIK. 190302037

Anggota Tim Penguji

Alva Hendi Muhammad, S.T., M.Eng., Ph.D.
NIK. 190302493

Hanafi, S.Kom., M.Eng., Ph.D.
NIK. 190302024

Pembimbing Pendamping

Dhani Ariatmanto, M.Kom., Ph.D
NIK. 190302197

Prof. Dr. Emo Utami, S.Si., M.Kom.
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 3 Juli 2023
Direktur Program Pascasarjana

Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Anggi Muhammad Rifa'i
NIM : 21.51.2099
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Analisis Komparasi Untuk Diagnosa Pneumonia Berdasarkan Hasil Citra Chest X-Ray Menggunakan Model ResNet-50 Dan MobileNetV2

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom.
Dosen Pembimbing Pendamping : Dhani Ariatmanto, M.Kom., Ph.D

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 3 Juli 2023
Yang Menyatakan,

Anggi Muhammad Rifa'i

HALAMAN PERSEMBAHAN

Pertama-tama puji syukur saya panjatkan pada Allah SWT atas terselesaikannya thesis ini dengan baik dan lancar.

Untuk mamah dan apa tercinta, terimakasih telah meluangkan banyak waktu untuk mendidik dan memberikan banyak ilmu yang tak terhingga, banyak hal yang tak bisa ku ceritakan dengan kata kata. Karena aku menangis sampai menyayat hati ketika aku mengingat perjuangan engkau yang telah banyak melalui rasa lelah dan sakit tetapi engkau hiraukan itu semua untuk menjalankan ibadah engkau, salah satunya memberikan pendidikan terbaik untuk ku. Terimakasih banyak, maafkan aku yang tidak bisa membalas semua perjuangan engkau semoga Allah subhanahu wa ta'ala meridhoi segala kebaikan yang telah engkau lakukan. Semoga do'a dan pencapaian ini menjadi persembahan istimewa saya untuk mamah dan apa. Aku ingin melakukan yang terbaik dari setiap kepercayaan engkau kepada ku. Terimakasih atas doa-doa kepada allah subhanahu wa ta'ala yang telah engkau panjatkan untuk ku. Aku sayang mamah apa, ridha Allah subhanahu wa ta'ala itu adalah karena ridha orangtua.

"Dan Kami perintahkan kepada manusia (berbuat baik) kepada dua orang ibu-bapanya; ibunya telah mengandungnya dalam keadaan lemah yang bertambah-tambah, dan menyapihnya dalam dua tahun. Bersyukurlah kepadaKu dan kepada dua orang ibu bapakmu, hanya kepada-Kulah kembalimu." (Qs. Luqman: 14)

Untuk adik ku tercinta, terimakasih atas do'a yang telah engkau berikan, walau berbisik dalam hati tetapi tedengar sampai menembus langit. Semoga menjadi anak yang sholehah dan bermanfaat untuk agama, orangtua dan negara.

HALAMAN MOTTO

Aku tidak peduli atas keadaan susah dan senangku, karena aku tidak tahu manakah di antara keduanya itu yang lebih baik bagiku.

(Umar bin Khattab)

Semuanya berjalan dengan rasa malas sepanjang jalan ini, kecuali hanya mereka yang diberi tahu rahasia-rahasia tindakan ilahi.

(Jalaluddin Rumi)

Salah satu pengkerdilan terkejam dalam hidup adalah membiarkan pikiran yang cemerlang menjadi budak bagi tubuh yang malas, yang mendahulukan istirahat sebelum lelah.

(Buya Hamka)

Jangan takut jatuh, kerana yang tidak pernah memanjatlah yang tidak pernah jatuh. Jangan takut gagal, kerana yang tidak pernah gagal hanyalah orang-orang yang tidak pernah melangkah. Jangan takut salah, kerana dengan kesalahan yang pertama kita dapat menambah pengetahuan untuk mencari jalan yang benar pada langkah yang kedua.

(Buya Hamka)

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Allhamdulillah, segala puji bagi Allah SWT atas segala limpahan ridho, hidayah, dan inayahnya sehingga thesis dengan judul "Analisis Komparasi Untuk Diagnosa Pneumonia Berdasarkan Hasil Citra Chest X-Ray Menggunakan Model ResNet-50 Dan MobileNetV2" ini dapat penulis selesaikan dengan baik dan lancar. Shalawat serta salam tetap tercurah untuk nabi besar, Muhammad SAW yang telah menunjukkan kepada kita dari zaman kegelapan ke zaman yang terang-benderang yaitu Dinul Islam.

Thesis ini disusun untuk memenuhi persyaratan memperoleh gelar Magister Komputer Universitas Amikom Yogyakarta. Dengan segala keterbatasan yang penulis miliki, masih banyak kekurangan-kekurangan yang harus diperbaiki. Semoga hasil penelitian ini dapat berguna, khususnya bagi dunia pendidikan. Dalam penulisan thesis ini, penulis banyak mendapat bantuan dari berbagai pihak. Oleh karena itu, ucapan terima kasih penulis sampaikan kepada:

- Bapak Prof. Dr. M. Suyanto, M.M. selaku Rektor Universitas AMIKOM Yogyakarta.
- Ibu Prof. Dr. Kusrini, M.Kom. selaku Direktur Program Pascasarjana Universitas AMIKOM Yogyakarta.
- Ibu Prof. Dr. Ema Utami, S.Si., M.Kom. selaku Wakil Direktur Program Pascasarjana Universitas AMIKOM Yogyakarta sekaligus selaku Pembimbing Utama.
- Bapak Dhani Ariatmanto, M.Kom., Ph.D selaku dosen Pembimbing Pendamping.
- Seluruh Dewan Penguji yang telah memberikan banyak masukan baik pada SPT, SHPT maupun UT untuk memperbaiki persiapan, proses dan pelaporan hasil penelitian ini.
- Ayahanda dan Ibunda tercinta yang dengan penuh kesabaran dan pengorbanannya selalu memberikan dorongan, bantuan material maupun non material agar penulis dapat menyelesaikan studi.

- Semua pihak yang tidak bisa penulis sebutkan satu persatu, terimakasih atas bantuan dan dukungannya.

Penulis menyadari thesis ini masih jauh dari sempurna, karena hal tersebut tidak lepas dari kelemahan dan keterbatasan penulis. Akhirnya penulis berharap agar thesis ini berguna sebagai tambahan ilmu pengetahuan serta dapat memberikan manfaat bagi semua pihak dan dijadikan implikasi selanjutnya bagi mahasiswa.

Billahi Fii Sabilil Haq, Fastabiqul Khairat

Yogyakarta, 3 Juli 2023

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR ISTILAH

Dataset: kumpulan data yang digunakan dalam penelitian di bidang data science, machine learning, dan deep learning. Ada dataset publik yang bebas diakses oleh siapa pun, serta dataset privat yang memiliki batasan akses.

Epoch: satu siklus latihan di mana model kecerdasan buatan diperbarui dengan data yang sama. Setiap epoch, bobot atau parameter pada neuron diperbaharui agar output model mendekati nilai yang diharapkan berdasarkan label data.

INTISARI

Penelitian ini melakukan analisis kinerja arsitektur ResNet50 dan MobileNetV2 dengan menerapkan teknik white balance dan CLAHE pada dataset yang digunakan. Tujuan penelitian ini adalah untuk menguji pengaruh teknik white balance dan CLAHE terhadap akurasi pengenalan kelas pada kedua arsitektur tersebut. Variabel penelitian yang dibatasi adalah jumlah kelas, jumlah epoch, dan waktu yang dibutuhkan per iterasi. Analisis dilakukan dengan menghitung akurasi, precision, recall, dan F1-Score dari hasil pengenalan kelas.

Hasil penelitian menunjukkan bahwa penggunaan ResNet50 dengan white balance dan CLAHE menghasilkan akurasi yang meningkat seiring dengan peningkatan jumlah epoch. Pada dataset dengan 2 kelas, ResNet50 menggunakan CLAHE dan White Balance mencapai akurasi sebesar 91,62% setelah 50 epoch, dengan precision, recall, dan F1-Score masing-masing sebesar 92,12%. Namun, pada dataset dengan 3 kelas, performa ResNet50 sedikit lebih rendah dengan akurasi 73,16%. Di sisi lain, penggunaan MobileNetV2 dengan white balance dan CLAHE menghasilkan hasil yang lebih baik. Pada dataset dengan 2 kelas, MobileNetV2 mencapai akurasi sekitar 99,76% setelah 50 epoch, dengan precision, recall, dan F1-Score mencapai 99,35%. Pada dataset dengan 3 kelas, MobileNetV2 mencapai akurasi sekitar 91,17% setelah 50 epoch, dengan precision, recall, dan F1-Score masing-masing sebesar 91,05%. Waktu yang dibutuhkan per iterasi untuk kedua arsitektur ini juga tergolong cepat, sekitar 21-23ms.

ResNet50 dan MobileNetV2 dengan white balance dan CLAHE meningkatkan akurasi pengenalan kelas pada dataset gambar. MobileNetV2 lebih unggul dalam akurasi pengenalan kelas, terutama pada dataset dengan 2 kelas. Namun, ResNet50 juga memberikan hasil yang baik terutama pada dataset dengan 2 kelas. Kedua arsitektur ini menunjukkan potensi untuk aplikasi pengenalan image classification dengan waktu pemrosesan yang cepat.

Kata kunci: MobileNetV2, ResNet50, White Balance, CLAHE, Pneumonia

ABSTRACT

This study analyzes the performance of the ResNet50 and MobileNetV2 architectures by applying white balance and CLAHE techniques to the dataset used. The purpose of this study was to examine the effect of white balance and CLAHE techniques on class recognition accuracy in both architectures. The limited research variables are the number of classes, the number of epochs, and the time required per iteration. The analysis was carried out by calculating the accuracy, precision, recall, and F1-Score from class recognition results.

The results show that using ResNet50 with white balance and CLAHE results in increased accuracy as the number of epochs increases. In a dataset with 2 classes, ResNet50 using CLAHE and White Balance achieves an accuracy of 91.62% after 50 epochs, with precision, recall and F1-Score each of 92.12%. However, on a dataset with 3 classes, ResNet50's performance is slightly lower with an accuracy of 73.16%. On the other hand, using MobileNetV2 with white balance and CLAHE produced better results. In a dataset with 2 classes, MobileNetV2 achieves an accuracy of around 99.76% after 50 epochs, with precision, recall and F1-Score reaching 99.35%. In a dataset with 3 classes, MobileNetV2 achieves an accuracy of around 91.17% after 50 epochs, with precision, recall and F1-Score each of 91.05%. The time required per iteration for these two architectures is also relatively fast, around 21-23ms.

ResNet50 and MobileNetV2 with white balance and CLAHE improve class recognition accuracy in image datasets. MobileNetV2 is superior in class recognition accuracy, especially on datasets with 2 classes. However, ResNet50 also gives good results especially on datasets with 2 classes. Both of these architectures show potential for image classification recognition applications with fast processing times.

Keyword: MobileNetV2, ResNet50, White Balance, CLAHE, Pneumonia

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Kesehatan merupakan salah satu unsur dasar kesejahteraan dalam memperbaiki tingkat sosial ekonomi masyarakat (Safira et al., 2019). Sebagai poin utama produktifitas bertahan hidup, manusia membutuhkan metabolisme serta ketahanan tubuh yang baik agar terhindar dari segala serangan penyakit maupun virus (Ahsan et al., 2020). Salah satu penyakit yang masih menjadi masalah yang serius tidak hanya di Indonesia tetapi di Dunia salah satunya adalah penyakit paru-paru, dalam penelitian yang dilakukan oleh (Yopento et al., 2022) menyatakan penyakit paru-paru merupakan penyakit yang serius dan dapat berakibat fatal bila tidak ditangani secara tepat, salah satu penyakit yang terjadi pada paru-paru adalah penyakit pneumonia. Pada penelitian yang dilakukan oleh (Estomihi, 2019) menyatakan pneumonia merupakan peradangan yang terjadi pada jaringan parenkim paru-paru yang sebagian besar disebabkan oleh mikroorganisme patogen dan sebagian kecil disebabkan oleh hal lain, keadaan ini paru-paru dipenuhi oleh cairan atau lendir.

Dalam penelitian lainnya yang dilakukan oleh (Çınar et al., 2020) menjelaskan Pneumonia merupakan penyakit infeksi penting dengan morbiditas dan mortalitas yang tinggi dan merupakan penyebab kematian yang umum di seluruh dunia, terutama pada anak di bawah 5 tahun dan 15% kematian disebabkan oleh pneumonia. Hasil dari laporan yang dilakukan oleh (Riskesdas, 2019)

Pneumonia masih menjadi penyebab tertinggi kematian balita maupun bayi baru lahir dan data dari riset kesehatan dasar (Riskesdas) 2018 di Indonesia Pneumonia merupakan penyebab kematian balita ke-2 di Indonesia setelah diare. Sehingga diagnosis dini pneumonia bisa memberikan penanganan lebih cepat yang bisa membantu para ahli dalam melakukan tindakan dengan segera dan metode pencitraan dalam diagnosis pneumonia yang paling banyak dipakai untuk melakukan diagnosis adalah Chest X-Ray (CXR) (Daniel et al., 2023).

Menutut Hamet & Tremblay dalam penelitian yang dilakukan oleh (Ligueran et al., 2022), kemajuan teknologi computer vision telah menyebabkan beberapa tonggak penting di bidang artificial intelligence, contoh implementasi teknologi computer vision telah diterapkan ke berbagai platform seperti situs jejaring sosial, jaringan kamera televisi sirkuit tertutup, dan robotika. Baru-baru ini, karena meningkatnya minat dalam pengembangan teknologi diagnostik medis berbantuan artificial intelligence dan informatika kesehatan/biomedis, computer vision perlahan tetapi bertahap digunakan dalam diagnosis penyakit (Nazir et al., 2023).

CXR sangat berguna dalam diagnosis penyakit tertentu dengan menggunakan computer bisa melakukan deteksi dan analisis penyakit ini secara teoritis dapat menghasilkan akurasi dan keandalan yang lebih tinggi dalam diagnosis medis, ada berbagai algoritma untuk implementasi teknologi computer vision yang salah satunya adalah Convolutional Neural Network (CNN) dengan jenis jaringan saraf tiruan (Ghose et al., 2022). Jaringan saraf tiruan adalah implementasi perangkat keras atau perangkat lunak yang meniru struktur dan

operasi jaringan saraf organik yang ada di otak manusia, jaringan saraf organik terdiri dari neuron dan koneksi dendrit-akson yang sesuai satu sama lain (Abraham & Nair, 2020).

CNN dijelaskan oleh (Venkatesan & Li, 2018) algoritma Deep Learning yang dapat mengambil gambar input, menetapkan keperluan (bobot dan bias yang dapat dipelajari) untuk berbagai aspek/objek dalam gambar dan dapat membedakan satu dari yang lain, CNN memiliki fitur utamanya yaitu kombinasi convolutional layer, ekstraksi bentuk visual, multilayer perceptron, realisasi pengenalan sesuai convolutional. Penelitian yang dilakukan oleh (Naveen & Diwan, 2021) mengenai penggunaan CXR dalam melakukan deteksi pneumonia dengan menggunakan model arsitek visual geometry group 19 (VGG-19) dari CNN untuk mengklasifikasikan menderita pneumonia atau normal, yang dapat membantu dokter dalam fase pengambilan keputusan ini mendapatkan nilai akurasi pengujian 95,67% dan 96% AUC pada dataset dengan 12,64 test loss dan 50 persen akurasi pengujian.

Dalam penelitian lainnya yang dilakukan (Daoud et al., 2021) mengkaji penerapan dua model CNN yang telah dilatih sebelumnya, yaitu AlexNet dan ResNet-50, menggunakan transfer learning untuk mengklasifikasikan CXR sebagai normal, pneumonia, dan COVID-19. Implementasi penelitian yang dilakukan oleh Daoud dengan model AlexNet pada fase training yang dikombinasikan dengan pendekatan klasifikasi hierarki memperoleh spesifisitas klasifikasi rata-rata makro, sensitivitas, dan skor F1 masing-masing sebesar 98,3%, 89,1%, dan 91,4%. Pada tahap selanjutnya dalam penelitian daoud, model ResNet-50 dalam fase training

yang dikombinasikan dengan pendekatan klasifikasi hierarkis mencapai spesifisitas, sensitivitas, dan skor F1 rata-rata makro masing-masing sebesar 97,4%, 95,2%, dan 94,9%.

Penelitian (Kolonne et al., 2021) ini memberikan pendekatan berbasis deep learning untuk mengklasifikasikan CXR sebagai Pneumonia, COVID-19, atau Normal. Metodologi dalam penelitian kolonne didasarkan pada model MobileNetV2 dengan layer tambahan yang ditambahkan ke bagian atas arsitektur dan telah dilatih dengan dan tanpa transfer learning, model tanpa transfer learning menunjukkan kinerja yang lebih tinggi, pendekatan tersebut menunjukkan rata-rata nilai akurasi, recall, precision dan F1-score masing-masing sebesar 98,65%, 98,15%, 98,20% dan 98,17%, dengan validasi silang 5 kali lipat, penelitian ini dapat diperluas dengan pengoptimal khusus yang dapat digunakan di lingkungan dengan sumber daya terbatas.

Sesuai dengan penjelasan dan permasalahan diatas, salah satu tindakan untuk mengetahui pasien terjangkit pneumonia adalah dengan melihat CXR pasien. Sejauh ini para tenaga medis melakukan analisa secara langsung dengan melihat hasil CXR pasien tanpa menggunakan sistem yang terkomputasi, maka menyebabkan permasalahan akurasi yang kurang baik dan subjektif. Perbedaan persepsi antar dokter spesialis paru dapat menyebabkan hasil diagnosa yang berbeda, Sehingga dibutuhkan teknologi yang dapat membantu dokter spesialis paru untuk menganalisa foto rontgen dengan cepat dan akurat dengan penggunaan teknologi informasi berbasis komputer dan data.

Untuk itu pada penelitian ini akan dilakukan pendeteksian pneumonia secara terkomputasi menggunakan CNN dengan membandingkan MobileNetV2 dan ResNet50 untuk melakukan klasifikasi penyakit kedalam empat kelas, yaitu: Bakteri Pneumonia, virus Pneumonia, dan Normal. Durasi time compsumnption juga menjadi fokus pada penelitian ini dengan menentukan apakah MobileNetV2 mengungguli ResNet50 dalam kecepatan waktu komputasi dalam hitungan time per step. Peneliti juga menambahkan tahap pre-processing yaitu white balance dan CLAHE pada model ResNet50 dan MobileNetV2 yang bertujuan untuk memperbaiki kualitas gambar dataset sehingga dapat meningkatkan akurasi pada model. Penulis berharap dengan adanya sistem ini dapat membantu meningkatkan hasil Analisa dokter spesialis paru dalam mendiagnosa penyakit paru, dan diharapkan akan memberikan kontribusi dalam penelitian dibidang Computer Vision.

1.2. Rumusan Masalah

Berdasarkan penjelasan pada latar belakang masalah, maka rumusan masalah pada penelitian ini adalah sebagai berikut.

a. Apakah arsitektur MobileNetV2 memberikan performa yang lebih baik dibandingkan dengan ResNet50 dalam mengklasifikasikan penyakit pneumonia pada dataset CXR?
b. Bagaimana perbandingan time consumption per-step antara arsitektur ResNet50 dan MobileNetV2 dalam melakukan klasifikasi pada dataset CXR?

c. Bagaimana penerapan algoritma white balance dan Contrast Limited Adaptive Histogram Equalization (CLAHE) dalam mempengaruhi performa arsitektur CNN dalam klasifikasi penyakit pneumonia pada dataset CXR?

1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut.

a. Penelitian ini melakukan analisis pada CXR Image dengan dua skema dataset, skema pertama menggunakan 4500 dataset yang terbagi menjadi tiga kelas yaitu 1500 normal CXR, 1500 bacterial pneumonia CXR, dan 1500 virus pneumonia CXR. Skema kedua menggunakan 3000 dataset yang terbagi menjadi dua kelas yaitu 1500 normal CXR dan 1500 pneumonia CXR.
b. Dataset yang digunakan bersumber dari University of California San Diego dipublish pada website Mendeley Data.
c. Pada penelitian ini penulis menggunakan platform Google Colaboratory dan Google Colaboratory Pro
d. Menggunakan System Linux, Python version 3.10.11, VGA Nvidia A100 SXM4 40GB, RAM 83.48.
e. Nilai epoch pada setiap training dengan interval 10, 20, dan 50 dari setiap skema arsitektur ResNet50 dan MobileNetV2 dalam melakukan klasifikasi penyakit pneumonia.
f. Setiap skema penelitian menggunakan base learning rate dengan nilai 0.001 dan menggunakan optimizer Adam

g. Penelitian ini hanya berfokus pada hasil klasifikasi penyakit pneumonia terbagi dua skema klasifikasi, skema pertama menjadi tiga klasifikasi yaitu, normal, bacterial pneumonia, dan virus pneumonia. Skema kedua menjadi dua klasifikasi yaitu normal dan pneumonia.

h. Indikator yang dipakai untuk membandingkan hasil performa model ResNet50 dan MobileNetV2 dari CNN menggunakan nilai accuracy, precision, recall, F1-score, dan time consumtion.

1.4. Tujuan Penelitian

Tujuan dari penelitian yang dilakukan adalah sebagai berikut:

a. Membandingkan performa arsitektur MobileNetV2 dengan ResNet50 dalam klasifikasi penyakit pneumonia pada dataset CXR dalam dua kelas dan tiga kelas dengan indikator nilai accuracy, precision, recall, F1-Score.

b. Menganalisis perbedaan waktu per-step antara arsitektur dalam melakukan klasifikasi pada dataset CXR, serta mengidentifikasi keunggulan MobileNetV2 dalam hal waktu yang lebih singkat dan konsisten.

c. Meneliti dampak penerapan algoritma white balance dan CLAHE pada performa metode CNN dalam mengklasifikasikan penyakit pneumonia pada dataset CXR.

1.5. Manfaat Penelitian

Bagian ini memuat penjelasan tentang:

a. Mampu melakukan analisis dan klasifikasi mengenai pneumonia dengan metode CNN kedalam dua skema, skema pertama dengan tiga kelas yaitu

normal, bacterial pneumonia, dan virus pneumonia. Skema kedua dengan dua kelas yaitu normal dan pneumonia.

b. Mengetahui pengaruh peningkatan dari model ResNet50 dan MobileNetV2 terhadap analisis klasifikasi pneumonia pada gambar CXR menggunakan CNN.

c. Menjadi referensi terhadap penelitian dimasa mendatang yang berkaitan dengan pemanfaatan metode ResNet50 dan MobileNetV2 pada gambar CXR menggunakan CNN.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Dalam penelitian ini penulis mencari informasi dari penelitian-penelitian sebelumnya sebagai perbandingan, baik mengenai kekurangan atau kelebihan penelitian yang sudah ada dan untuk dijadikan sebagai bahan kajian dalam mengetahui keterkaitan antara penelitian terdahulu dengan penelitian yang penulis lakukan dan untuk menghindari terjadinya duplikasi yang dilakukan sehingga penelitian sebelumnya yang sudah dilakukan sangatlah penting bagi penulis. Tinjauan pustaka juga bertujuan untuk menunjukkan bahwa penelitian yang penulis lakukan sangatlah bermanfaat dan memiliki arti penting sebagai kontribusi penelitian terhadap ilmu pengetahuan. Berikut ini adalah beberapa ulasan jurnal tentang penelitian terdahulu berkenaan dengan data dan metode yang digunakan penulis sebagai acuan.

Berdasarkan penelitian sebelumnya terkait dengan tersedianya dataset CXR pneumonia yang dapat diakses oleh publik sehingga penulis tertarik untuk melakukan penelitian terkait dengan ide tentang memanfaatkan CXR image untuk melakukan identifikasi penyakit pneumonia berdasarkan infeksi bakterinya. Selanjutnya untuk proses CNN penulis menggunakan metode yang telah dikaji berdasarkan penelitian sebelumnya tentang penggunaan metode tersebut Berikut beberapa kajian tentang penggunaan dengan menerapkan arsitektur ResNet50 dan MobileNetV2.

Penelitian (Roy et al., 2022) mengusulkan metode Singular Value Decomposition dengan Contrast Limited Adaptive Histogram Equalization (SVD-CLAHE) Boosting untuk mendeteksi penyakit Covid-19 secara efisien. Peneliti menggunakan data set gambar CXR yang dikumpulkan secara personal. Peneliti menggunakan metode novel loss function yaitu Balanced Weighted Categorical Cross-Entropy (BWCCE) untuk meningkatkan tingkat akurasi. Dari hasil eksperimen, performa metode dengan model ResNet50 pada kumpulan data yang diperbesar menggunakan SVD-CLAHE Boosting, bersama dengan fungsi loss BWCCE, mencapai F1 score 95%, accuracy 94%, recall 96%, dan precision 96%, yang jauh lebih baik dibandingkan hasil model CNN konvensional lainnya seperti InceptionV3, DenseNet-121, Xception, dll. Namun, dalam penelitian (Roy et al., 2022) belum menjelaskan terkait penelitian pada penyakit pneumonia sedangkan jenis dataset yang digunakan sama.

Selanjutnya penelitian yang dilakukan oleh (Kumar Sethy et al., 2020) menggunakan metodologi berbasis deep learning untuk mendeteksi pasien terinfeksi virus corona menggunakan gambar CXR. Support vector machine (SVM) mengklasifikasikan gambar CXR yang terkena corona virus dari orang lain menggunakan deep feature, metodologi ini bermanfaat bagi praktisi medis untuk melakukan diagnosis pasien yang terinfeksi virus corona. Model klasifikasi yang disarankan dari penelitian Kumar Sethy yaitu ResNet50 dan SVM mencapai accuracy, FPR, F1 score, MCC and Kappa masing-masing adalah 95,38%, 95,52%, 91,41% dan 90,76% untuk mendeteksi Covid-19 (mengabaikan SARS, MERS dan ARDS). Akan tetapi dalam penelitian Kumar Sethy belum melakukan penelitian

menggunakan model arsitektur MobileNetV2 yang memiliki time consumtion yang lebih cepat dan belum menjelaskan terkait klasifikasi penyakit pneumonia sedangkan jenis dataset yang digunakan sama.

Arsitektur ResNet50 juga digunakan pada penelitian yang dilakukan oleh (Shadab et al., 2022) untuk menentukan apakah kanker atau tidak dengan menggunakan teknik machine learning dan untuk melakukan klasifikasi dengan tahap training CNN menggunakan Arsitektur ResNet50. Hasil penelitian ini mendapatkan akurasi maksimum 91,7% dalam seluruh proses deteksi, ini difokuskan pada proses deteksi dengan melakukan training pada neural network dan mendapatkan hasil terbaik sehingga deteksi dini kanker dapat dilakukan. Dengan acuan hasil terbaik penelitian yang dilakukan oleh shadab, maka penulis menyarankan untuk melakukan implementasi arsitektur ResNet50 pada dataset CXR.

Penelitian lainnya yang dilakukan oleh (Lu et al., 2020) menggunakan ResNet dengan medical imaging untuk diagnosis penyakit otak dengan memberikan kesimpulan yang jelas tentang otak bagian dalam. Dalam penelitian yang dilakukan Lu, sistem deteksi otak patologis baru diusulkan menggunakan gambar resonansi magnetic atau X-Ray otak berdasarkan ResNet dan randomized neural networks, hasil pada sistem pengembangan yang dilakukan Lu mencapai akurasi rata-rata terbaik sebesar 95,00% dalam membedakan otak patologis dari kontrol normal. Namun, penelitian yang dilakukan Lu menggunakan dataset X-Ray otak dan bisa dikembangkan menggunakan dataset CXR, sehingga bisa bermanfaat dalam bidang diagnosis lainnya.

ResNet50 juga digunakan dalam penelitian (Liu et al., 2022) bertujuan untuk melakukan deteksi Covid-19 secara efektif, jaringan multiscale class residual attention (MCRA) diusulkan melalui klasifikasi gambar CXR. Untuk menghasilkan spatial attention untuk setiap class, yang dapat menghindari interferensi inter-class dan meningkatkan fitur terkait untuk lebih meningkatkan deteksi COVID-19, Sehingga mendapatkan hasil eksperimen bahwa dapat mencapai kinerja diagnostik yang unggul dengan menggunakan ResNet50 pada dataset COVIDx, dan accuracy, PPV, sensitivity, specificity and F1-score masing-masing adalah 97,71%, 96,76%, 96,56%, 98,96%, dan 96,64%. Namun, dalam penelitian Liu belum menjelaskan pada penyakit pneumonia sedangkan jenis dataset yang digunakan sama dan dapat melakukan modifikasi alur dalam penelitian yang telah dilakukan Liu.

Selanjutnya penelitian (Emin Sahin, 2022) mengusulkan model CNN untuk identifikasi Covid-19 otomatis menggunakan gambar CXR, model CNN yang diusulkan dirancang untuk menjadi alat diagnostik yang andal untuk kategorisasi dua kelas (Covid-19 dan Normal). Arsitektur yang diterapkan juga berbeda yaitu MobileNetV2 dan ResNet50 yang telah masuk tahap training sebelumnya, dievaluasi untuk kumpulan data Covid-19 ini dengan jumlah 13.824 gambar CXR dan model yang kami sarankan dibandingkan dengan algoritme deteksi Covid-19 yang ada di istilah akurasi. Hasil eksperimen yang dilakukan Emin menunjukkan bahwa model yang diusulkan untuk melakukan identifikasi pasien dengan penyakit Covid-19 dengan accuracy 96,71%, F1-score 91,89% sehingga model ini dapat membantu dokter dalam membuat penilaian yang tepat tentang cara mendiagnosis Covid-19. Namun, penelitian Emin masih bisa dikembangkan karena mengunakan

dataset CXR dengan pengembangan untuk melakukan diagnosis penyakit pneumonia.

Penelitian lainnya yang dilakukan oleh (Dilshad et al., 2021) menggunakan MobileNetV2, peneliti menggunakan teknik Artificial Intelligence yang disebut CNN dengan menggunakan arsitektur MobileNetV2 dikarenakan lebih cepat daripada model convolutional biasa dan secara substansial mengurangi biaya komputasi. Dataset yang digunakan pada penelitian Dilshad terdiri dari 447 dan 447 gambar CXR Covid-19 dan Nofindings, total 894 gambar CXR, setelah itu dibagi menjadi 4 bagian yaitu training, validation, testing dan local/aligarh dataset. Hasil eksperimen yang dilakukan oleh Dilshad menunjukkan accuracy 96,33%, F1-score adalah 93% dan 96% pada pengujian dataset pertama dan pengujian kedua (local/aligarh), recall/sensitivity dari classifier adalah 93% dan 96% untuk pengujian dataset pertama dan pengujian kedua (local/aligarh). Pengembangan yang bisa dilakukan dari penelitian Dilshad dengan menambahkan jumlah dataset yang akan diuji dan presentasi pembagian dataset training, validation, testing dan local/aligarh serta melakukan menambahkan perbandingan arsitektur lainnya.

2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian

Analisis Komparasi Untuk Diagnosa Pneumonia Berdasarkan Hasil Citra Chest X-Ray Menggunakan Model ResNet-50 dan MobileNetV2

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Classification of lungs infected Covid-19 images based on inception-ResNet	Yunfeng Chen, Yalan Lin, Xiaodie Xu, Jinzhen Ding, Chuzhao Li, Yiming Zeng, Weili Liu, Weifang Xie, Jianlong Huang Computer Methods and Programs in Biomedicine, 2022	Melakukan diagnosis pasien Covid-19 dini dan parah dan juga dapat diterapkan pada klasifikasi patologis dan prediksi prognosis lainnya.	Dengan penggunaan citra CT dalam diagnosis klinis citra COVID-19 secara luas dan jumlah sampel yang diterapkan terus meningkat, metode dalam penelitian ini diharapkan dapat menjadi alat diagnostik tambahan yang dapat secara efektif meningkatkan akurasi diagnostik citra klinis COVID-19. Dengan hasil nilai pengukuran statistik yang diperoleh Inception-ResNet adalah 88,23%, 83,45%, 89,72%, 95,53% dan 88,74%.	Menggunakan beberapa dataset yang memiliki kualitas citra yang bagus dan berbeda sehingga bisa mengetahui berapa persen pengaruh dari dataset. Membandingkan dengan dataset dari CXR sehingga bisa mengetahui mana yang nilai statistik yang lebih tinggi.	Menerapkan arsitektur ResNet50 dan MobileNetV2 untuk melakukan analisis penyakit pneumonia.
2	Using handpicked features in	Sheetal Rajpal, Navin Lakhyani,	Karena pneumonia dan Covid-19 gejala	Module pertama menggunakan transfer	Melakukan perbandingan dengan	Melakukan perbandingan arsitektur dengan

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	conjunction with ResNet-50 for improved detection of Covid-19 from chest X-ray images	Ayush Kumar Singh, Rishav Kohli, Naveen Kumar Chaos, Solitons and Fractals, 2021	yang serupa/ overlapping symptoms dan mempengaruhi paru-paru manusia. Dengan tujuan melakukan klasifikasi foto citra X-Ray pasien pneumonia dan pasien Covid-19 dengan mengusulkan framework baru yang menggabungkan dari 3 module.	learning dari ResNet50 untuk melakukan training pada neural network pada pre-processing dan mendapatkan 2048 feature, Module kedua merancang kumpulan 252 feature pilihan berbasis frekuensi dan tekstur yang direduksi menjadi 64 feature menggunakan PCA. Module ketiga menggabungkan feature yang dihasilkan dari module pertama dan module kedua lalu melanjutkannya ke dense layer diikuti oleh softmax layer untuk menghasilkan model klasifikasi. Model menghasilkan akurasi klasifikasi keseluruhan 0,974 ± 0,02 dan sensitivitas 0,987 ± 0,05,	arsitektur CNN yang berbeda sehingga bisa mengetahui mana arsitektur yang memiliki nilai yang tertinggi.	menggunakan ResNet50 dan MobileNetV2 untuk mengetahui mana yang memiliki nilai hasil yang tertinggi dan melakukan klasifikasi jenis-jenis dari penyakit pneumonia.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				0,963 ± 0,05, dan 0,973 ± 0,04 pada interval kepercayaan 95% untuk kelas COVID-19, normal, dan pneumonia.		
3	Efficacy of Transfer Learning-based ResNet models in Chest X-ray image classification for detecting Covid-19 Pneumonia	Sadia Showkat, Shaima Qureshi Chemometrics and Intelligent Laboratory Systems, 2022	Melakukan penelitian dengan menggunakan dari setiap generasi ResNet untuk melakukan deteksi Covid-19 dengan menggunakan hasil citra foto rontgen X-Ray.	Arsitektur ResNet terbukti efisien ResNet untuk melakukan deteksi Covid-19 dengan menggunakan hasil citra foto rontgen X-Ray dengan mencapai tingkat akurasi lebih dari 90%.	Mencoba menerapkan dataset yang bervariatif dan dibandingkan dengan model arsitektur yang dikembangkan telah ditentukan.	Melakukan analisis dan klasifikasi dari jenis-jenis penyakit pneumonia dan menambahkan arsitektur MobileNetV2 pada metode CNN.
4	Gender classification on digital dental x-ray images using deep convolutional neural network	M.V. Rajee, C. Mythili Biomedical Signal Processing and Control, 2021	Untuk membantu forensic recognition maka melukukan penelitian klasifikasi gender berdasarkan digital dental x-ray dengan menggunakan deep learning CNN untuk membantu bidang.	Deep learning CNN yang menggunakan arsitektur ResNet50, VGG16, dan Alexnet. Dengan melakukan tiga tahap yaitu pre-processing, segmentation, dan klasifikasi gender. Dengan tingkat akurasi klasifikasi tertinggi 98,27% menggunakan arsitektur ResNet-50.	Menerapkan akrsitektur lainnya yang baru dan menerapkan dataset yang bervariatif sehingga mendapatkan hasil perbandingan dan peningkatan yang bervariatif.	Dalam proses pengolahan gambar menggunakan chest X-Ray, menggunakan artiketur CNN ResNet-50 dan MobileNetV2, menghasilkan analisis klasifikasi jenis-jenis penyebab penyakit pneumonia berdasarkan infeksi bakteri.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	Deep learning-based approach for detecting Covid-19 in chest X-rays	M. Emin Sahin Biomedical Signal Processing and Control, 2022	Melakukan diagnosis Covid-19 kedalam dua kelas yaitu normal dan Covid-19, dengan Menggunakan arsitektur yang berbeda yaitu MobileNetV2 dan ResNet50.	Hasil eksperiment terhadap identifikasi pasien pengidap peyakit Covid-19 mendapatkan nilai tertinggi akurasi 96,71% dan 91,89% untuk nilai F1-Score.	Menggunakan dataset yang berbeda dan bervariatif sehingga mendapatkan hasil perbandingan dan peningkatan yang bervariatif.	Melakukan klasifikasi dan diagnosa jenis penyakit pneumonia berdasarkan infeksi bakteri dengan acuan data menggunakan citra gambar X-Ray.
6	Covid-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches	Mesut Togaçar, Burhan Ergen, Zafer Cömert Computers in Biology and Medicine, 2020	Klasifikasi dataset X-Ray kedalam tiga kelas yaitu pneumonia, coronavirus, normal. Pada dataset yang dimiliki dilakukan proses rekontruksi menggunakan teknik fuzzy color dalam tahap pre-processing dan struktur gambar menggunakan ukuran asli. Menggunakan arsitektur dari MobileNetV2 dan SqueezeNet lalu	Dari keseluruhan proses klasifikasi mendapatkan nilai hasil akurasi 99,27% dengan melakukan kombinasi arsitektur MobileNetV2 dan SqueezeNet serta melakukan klasifikasi menggunakan SVM.	Komparasi arsitektur model CNN dilakukan sendiri sehingga bisa mendapatkan validasi dan apakah memiliki perbedaan dengan penelitian sebelumnya.	Pada prosesnya menggunakan arsitektur model CNN ResNet50 dan MobileNetV2 untuk melakukan klasifikasi jenis penyakit pneumonia berdasarkan infeksi bakteri.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			dilakukan tahap klasifikasi menggunakan Support Vector Machines (SVM).			
7	Automated image classification of chest X-rays of COVID-19 using deep transfer learning	Sara Dilshad, Nikhil Singh, M. Atif, Atif Hanif, Nafeesah Yaqub, W.A. Farooq, Hijaz Ahmad, Yu-ming Chu, Muhammad Tamoor Masood Results in Physics, 2021	Identifikasi lebih cepat infeksi Covid-19 dengan penggunaan computer yang memiliki spesifikasi rendah. Arsitektur MobileNetV2 dipilih karena model convolutional yang cepat dan biaya komputasi yang murah.	Hasil eksperimen model yang telah dilakukan dengan mendapatkan akurasi 96,33%, nilai klasifikasi recall/sensitivity adalah 93% dan 96% untuk testing pertama dan testing kedua (local/aligarh).	Melakukan test dengan menggunakan arsitektur yang lainnya sehingga data yang dihasilkan lebih variatif.	Melakukan klasifikasi dan diagnosa jenis penyakit pneumonia berdasarkan infeksi bakteri dengan acuan data menggunakan citra gambar X-Ray menggunakan arsitektur ResNet50 dan MobileNetV2.
8	Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep	Ali Narin, Ceren Kaya, Ziynet Pamuk Pattern Analysis and Applications, 2021	Implementasi sistem deteksi yang menjadi alternatif yang cepat dalam melakukan diagnosis penyakit Covid-19. Penelitian ini menggunakan arsitektur ResNet50,	Hasil penelitian ini telah di implementasikan dengan empat kelas yaitu Covid-19, viral pneumonia, bacterial pneumonia, normal. Dengan hasil preforma yang paling tinggi dari	Mencoba klasifikasi dan diagnosa jenis penyakit pneumonia berdasarkan infeksi bakteri.	Dalam proses klasifikasi terbagi 4 kelas yaitu bacterial pneumonia, viruses pneumonia, fungi pneumonia, normal. Pada prosenya juga menggunakan model CNN dengan arsitektur MobileNetV2.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	convolutional neural networks		ResNet101, ResNet152, InceptionV3 and Inception-ResNetV2 dengan menggunakan CXR.	arsitektur ResNet-50, pada dataset satu hasil akurasi 96,1%, untuk dataset dua 99,5% dan 99,7% pada dataset tiga.		
9	SVD-CLAHE boosting and balanced loss function for Covid-19 detection from an imbalanced Chest X-ray dataset	Santanu Roy, Mrinal Tyagi, Vibhuti Bansal, Vikas Jain Computers in Biology and Medicine, 2022	Mengusulkan teknik augmentasi data baru, yang disebut SVD-CLAHE (Singular Value Decomposition - Contrast Limited Adaptive Histogram Equalrzation) Boosting, dan novel loss function Balanced Weighted Categorical Cross-Entropy (BWCCE), untuk mendeteksi Covid 19 penyakit secara efisien dari set data gambar X-ray Dada (CXR) yang sangat tidak scimbang menggunakan	Metode Boosting SVD-CLAHE terdiri dari metode oversampling dan under-sampling. Hasil eksperimen mengungkapkan bahwa model ResNet-50 pada kumpulan data yang diperbesar (oleh SVD-CLAHE Boosting), bersama dengan lost function BWCCE, mencapai skor 95% F1 score, 94% accuracy, 96% recall, and 96% precision yang jauh lebih baik dibandingkan hasil model CNN konvensional lainnya seperti InceptionV3,	Melakukan pengujian dengan menggunakan arsitektur model CNN lainnya sehingga tingkat akurasi terbaik bisa dibuktikan dan divalidasi.	Menerapkan arsitektur model CNN MobileNetV2 yang digunakan untuk melakukan klasifikasi penyakit pneumonia.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			arsitektur ResNet-50.	DenseNet-121, Xception, dll.		
10	Detection of Coronavirus Disease (COVID-19) based on Deep Features and Support Vector Machine	Prabira Kumar Sethy, Santi Kumari Behera, Pradyumna Kumar Ratha, Preesat Biswas International Journal of Mathematical, Engineering and Management Sciences, 2020	Dalam penelitian ini, dilakukan dengan metodologi berbasis deep learning yang disarankan untuk mendeteksi pasien terinfeksi virus corona menggunakan gambar citra X-Ray. Support vector machine melakukan klasifikasi gambar X-Ray menggunakan deep feature ResNet-50	Model klasifikasi yang disarankan yaitu ResNet-50 dan SVM mencapai accuracy, FPR, F1 score, MCC and Kappa masing-masing adalah 95,38%, 95,52%, 91,41% dan 90,76% untuk mendeteksi COVID-19 sehingga arsitektur ResNet-50 dengan menambahkan SVM lebih unggul dari arsitektur lainnya.	Mencoba klasifikasi dan diagnosa jenis penyakit pneumonia berdasarkan infeksi bakteri.	Melakukan implementasi arsitektur model CNN MobileNetV2 dengan membagi kedalam empat kelas yaitu bacterial pneumonia, viruses pneumonia, fungi pneumonia, normal.
11	Detection of cancer from histopathology medical image data using ML with CNN ResNet-50 architecture	Shadan Alam, Shadab, M.A. Ansari, Nidhi Singh, Aditi Verma, Pragati Tripathi, Rajat Mehrotra	Penelitian yang dilakukan ialah mendeteksi tumor otak atau kanker untuk meningkatkan proses deteksi dini tumor otak dan kanker. Untuk menentukan apakah	Menggunakan dua jenis dataset, salah satunya adalah dataset Breast cancer Wisconsin dan dataset gambar Breast Histopathology (citra sel biopsi). Kami mendapatkan akurasi maksimum 91,7% dalam	Menggunakan deep learning, melakukan test dengan menggunakan arsitektur yang lainnya sehingga data yang dihasilkan lebih variatif.	Dalam proses pengolahan gambar menggunakan CXR dengan menambahkan arsitektur model CNN MobilenetV2 dan melakukan klasifikasi penyebab penyakit pneumonia berdasarkan infeksi bakteri.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
		Computational Intelligence in Healthcare Applications, 2022	suatu gambar bersifat kanker atau tidak, pada prosesnya menggunakan teknik machine learning. Untuk klasifikasi, melatih Arsitektur ResNet-50.	seluruh proses deteksi kami. Kami dapat menggunakan MRI, CT scan, atau semua jenis citra medis dalam implementasi.		
12	Detecting pathological brain via ResNet and randomized neural networks	Siyuan Lu, Shui-Hua Wang, Yu-Dong Zhang Heliyon, 2020	Diagnosis dengan bantuan komputer memainkan peran yang semakin penting di klinik, yang dapat membantu dokter menganalisis medical imaging secara otomatis. Dalam penelitian ini, sistem deteksi otak patologis baru diusulkan untuk gambar resonansi magnetik otak berdasarkan ResNet dan randomized neural networks.	Pertama, ResNet-50 digunakan sebagai ekstraktor fitur, yang merupakan struktur convolutional neural network yang terkenal. Kemudian, menggunakan tiga randomized neural networks, yaitu the schmidt neural network, the random vector functional-link net, dan the extreme learning machine. Bobot dan bias di tiga jaringan dilatih oleh the chaotic bat algorithm. Mencapai akurasi rata-rata terbaik	Melakukan test dengan menggunakan dataset yang berbeda sehingga data yang dihasilkan lebih variatif.	Menambahkan pengujian dengan arsitektur model CNN MobileNetV2 dan menggunakan dataset X-Ray untuk melakukan klasifikasi penyakit peumonia.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				sebesar 95.00% dalam membedakan otak patologis dari kontrol normal, yang sebanding dengan beberapa metode canggih.		
13	COVID-19 diagnosis via chest X-ray image classification based on multiscale class residual attention	Shangwang Liu, Tongbo Cai, Xiufang Tang, Yangyang Zhang, Changgeng Wang	bertujuan untuk mendeteksi COVID-19 secara efektif, jaringan multiscale class residual attention (MCRA) diusulkan melalui klasifikasi gambar X-Ray (CXR).	Hasil eksperimen menunjukkan bahwa mencapai kinerja diagnostik yang unggul dengan menggunakan ResNet-50 pada dataset COVIDx, dan accuracy, PPV, sensitivity, specificity and F1-score masing-masing adalah 97,71%, 96,76%, 96,56%, 98,96%, dan 96,64%.	Menambahkan arsitektur model CNN MobileNetV2 sehingga bisa mengetahui apakah bisa diimplementasikan kedalam device yang memiliki spesifikasi rendah.	Melakukan klasifikasi penyebab penyakit pneumonia berdasarkan infeksi bakteri dan menambahkan arsitektur model CNN MobileNetV2.

2.3. Landasan Teori

2.3.1. X-Ray

Pada buku (Rosenbusch & Eekelen, 2019) X-Ray ditemukan pada tahun 1895 oleh Wilhelm Röntgen, X-Ray adalah jenis radiasi elektromagnetik yang digunakan oleh dokter dan radiografer untuk mencari tahu lebih banyak tentang tubuh manusia. Dalam buku tersebut juga menjelaskan bahwa sejak saat itu telah menjadi alat diagnostik yang penting dalam bidang kedokteran. X-Ray dijelaskan pada buku (Collins & Stern, 2015) dapat digunakan untuk menemukan patologi atau masalah dengan tulang, otot, sendi, dll. X-Ray juga dapat digunakan untuk menemukan masalah dengan organ-organ dalam seperti jantung, paru-paru, dan ginjal.

(Rizal Makarim, 2022) menjelaskan bahwa X-Ray berfungsi dengan menghantarkan gelombang elektromagnetik melalui tubuh yang dianalisis. Ketika gelombang ini menabrak organ-organ atau struktur dalam tubuh, mereka memantulkan atau difraksikan. Selanjutnya rizal menyatakan gelombang ini kemudian ditangkap oleh detektor X-Ray dan ditranslasikan ke dalam citra yang ditampilkan pada layar monitor. Berdasarkan pada buku (Corne & Pointon, 2010) menyatakan citra ini kemudian dapat dianalisis oleh dokter untuk menemukan masalah dalam tubuh.

Tidak hanya digunakan sebagai alat diagnostik, dalam buku (Reed, 2011) menjelaskan bahwa X-Ray juga dapat digunakan untuk mengobati penyakit, radiasi X-Ray dapat digunakan untuk menghancurkan tumor atau sel-sel kanker. Dijelaskan pada buku (Balachandran, 2014) Aliran X-Ray yang tepat juga dapat

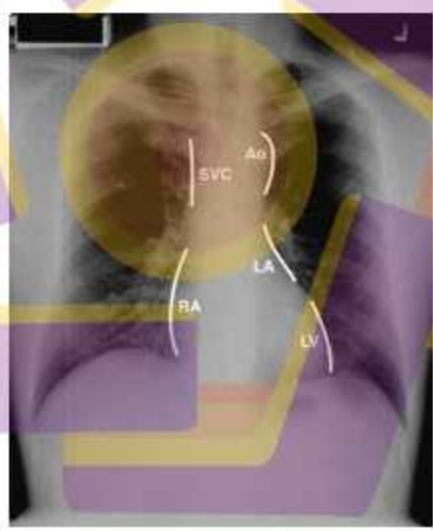
digunakan untuk mengobati penyakit seperti osteoporosis dengan menguatkan tulang. Meskipun X-Ray dapat membantu dengan diagnosis dan pengobatan seperti yang dijelaskan diatas, penting untuk mengingat bahwa X-Ray adalah jenis radiasi yang harus digunakan dengan hati-hati dan hanya ketika berada di bawah pengawasan dokter.

2.3.2. Pneumonia

Menurut (Eng & Cheah, 2005) dalam buku Interpreting Chest X-Rays mengatakan pneumonia atau yang sering disebut dengan penyakit paru-paru basah yang dapat menyebabkan inflamasi terhadap pundi-pundi udara didalam satu ataupun kedua belah paru-paru, CXR menampilkan bayangan fokus pada lobus kanan bawah dengan air bronchograms yang menunjukkan pneumonia. Selanjutnya penelitian (Tukbekova et al., 2019) menyatakan bahwa pneumonia dapat menyebabkan komplikasi serius, termasuk gagal napas dan infeksi lainnya, bakteri ini pertama kali terlihat pada saluran udara orang yang meninggal oleh Edwin Klebs pada tahun 1875.

Pembahasan dalam penelitian (Niederman et al., 2005) menyatakan bahwa pneumonia dapat disebabkan oleh berbagai jenis mikroorganisme, termasuk virus, bakteri, jamur dan parasit. Menurut penelitiannya, bakteri yang paling sering menyebabkan pneumonia adalah Streptococcus pneumoniae, Haemophilus influenzae, dan Staphylococcus aureus. Sedangkan virus yang paling sering menyebabkan pneumonia menurut penelitian Niederman adalah virus influenza, virus parainfluenza, virus respiratori sincitial, dan virus varicella-zoster. Jamur

yang paling sering menyebabkan pneumonia adalah Cryptococcus neoformans, Histoplasma capsulatum, dan Coccidioides immitis. Parasit yang paling sering menyebabkan pneumonia menurutnya adalah Pneumocystis jirovecii. Selain itu Niederman menyatakan bahwa faktor risiko lain yang dapat menyebabkan pneumonia termasuk merokok, menjalani terapi antibiotik, sistem imunitas yang lemah, usia lanjut, penyakit paru kronis, dan penyakit lainnya. Buku (Eng & Cheah, 2005) menyatakan untuk batas jantung pada PA CXR ditunjukkan pada gambar 2.1 SVC – vena cava superior, RA – atrium kanan, Ao – buku jari aorta, LA – atrium kiri, LV – ventrikel kiri.



Gambar 2.1 Ilustrasi X-Ray

Pneumonia memiliki banyak jenis yang klasifikasi berdasarkan dari mana mendapatkan infeksi tersebut dan yang akan peneliti bahas berfokus pada jenis *community-acquired pneumonia* atau pneumonia yang terinfeksi dari masyarakat

adalah jenis pneumonia yang paling umum. Itu terjadi di luar rumah sakit atau fasilitas perawatan kesehatan lainnya, ini mungkin disebabkan oleh (Torres et al., 2021):

- **Bacteria,** pneumonia jenis ini dapat terjadi dengan sendirinya atau setelah Anda mengalami pilek atau flu. Ini dapat mempengaruhi satu bagian (lobus) paru-paru, suatu kondisi yang disebut pneumonia lobar.
- **Fungi,** pneumonia jenis ini paling sering terjadi pada orang dengan masalah kesehatan kronis atau sistem kekebalan yang lemah, dan pada orang yang menghirup organisme dalam dosis besar. Jamur penyebabnya dapat ditemukan di tanah atau kotoran burung dan bervariasi tergantung lokasi geografis.
- **Viruses,** beberapa virus yang menyebabkan pilek dan flu dapat menyebabkan pneumonia. Virus adalah penyebab paling umum pneumonia pada anak di bawah 5 tahun. Pneumonia virus biasanya ringan, tetapi dalam beberapa kasus bisa menjadi sangat serius. Coronavirus 2019 (COVID-19) dapat menyebabkan pneumonia, yang dapat menjadi parah.

2.3.3. Computer Vision

Computer vision dalam buku yang dibuat oleh (Ballard & Brown, 1982) adalah bidang ilmu yang mempelajari cara komputer dapat memproses, menganalisis, dan memahami gambar dan video secara otomatis. Menurut Ballard juga tujuan utama dari computer vision adalah untuk mengembangkan algoritma dan teknologi yang memungkinkan komputer untuk "melihat" dunia seperti manusia. Contohnya termasuk pengenalan wajah, pengenalan objek, deteksi

gerakan, dan analisis citra medis. Menurut buku computer vision oleh (Szeliski, 2011) computer vision telah menjadi salah satu bidang paling aktif dalam pengembangan teknologi kecerdasan buatan dalam beberapa tahun terakhir. Hal ini terjadi karena kemajuan besar dalam pemrosesan gambar digital, pengenalan pola, dan pembelajaran mesin. Szeliski menyatakan bahwa teknologi computer vision telah digunakan dalam berbagai industri, termasuk perbankan, transportasi, manufaktur, dan kesehatan. Seperti pada gambar 2.2 computer vision telah digunakan dalam bidang kesehatan untuk melakukan klasifikasi pada Citra CXR dengan berbagai hasil dari diagnosis.

Gambar 2.2 Deteksi dan Pengenalan

2.3.4. Convolutional Neural Networks

Convolutional networks (LeCun & Bengio, 1995), juga dikenal sebagai CNN, adalah jenis jaringan saraf khusus yang digunakan dalam bidang computer vision. CNN berbeda dengan neural network tradisional karena memiliki lapisan konvolusi yang memungkinkan model untuk secara efektif mengidentifikasi fitur-fitur yang berkaitan dengan gambar. Menurut (LeCun et al., 2015), CNN telah menjadi teknologi yang sangat penting dalam pengolahan gambar digital dan telah

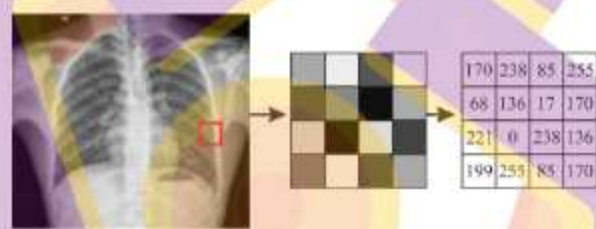
digunakan dalam berbagai aplikasi seperti pengenalan objek, deteksi wajah, dan klasifikasi citra medis. Disebutkan juga kemampuan CNN dalam memproses informasi gambar telah membuatnya menjadi teknologi yang sangat penting dalam bidang computer vision, keunggulan lain dari CNN adalah kemampuannya untuk belajar fitur-fitur yang relevan dari data gambar secara mandiri, sehingga model dapat belajar secara otomatis tanpa perlu pemrosesan manual dari manusia.

Dalam buku (Goodfellow et al., 2016) menurutnya jaringan convolutional telah sangat sukses dalam aplikasi praktis, nama "convolutional neural networks" menunjukkan bahwa jaringan menggunakan operasi matematika yang disebut konvolusi. Konvolusi adalah jenis khusus dari operasi linier, jaringan konvolusi hanyalah jaringan saraf yang menggunakan konvolusi sebagai pengganti perkalian matriks umum di setidaknya salah satu layer. Berdasarkan penelitian (Desai & Shah, 2021) arsitektur CNN secara umum terdiri dari feature learning (convolutional layer, activation layer ReLU dan pooling layer) serta structural condition classification (flatten, fully connected layer, dan Softmax) seperti pada gambar 2.3.

Gambar 2.3 Proses CNN

2.3.5. Convolutional Layer

Convolutional Layer yang dijelaskan oleh (Ke et al., 2018) adalah blok struktur utama CNN, berisi satu set filter atau kernel yang parameternya harus dipelajari selama training, ukuran filter biasanya lebih kecil dari gambar sebenarnya, setiap filter menyatu dengan gambar dan membuat activation map. Untuk convolution, filter melingkupi tinggi, lebar gambar, pixel diantara setiap elemen filter, dan input dihitung pada setiap posisi spasial. Berikut pada gambar 2.4 adalah ilustrasi bagaimana value pixel didapatkan, yaitu berdasarkan intensitas warna pada gambar.

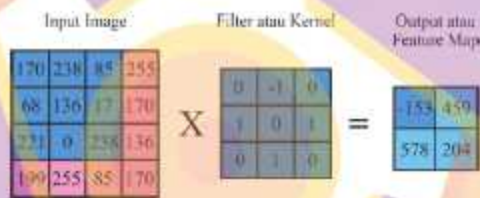


Gambar 2.4 Value Pixel Gambar

Pada gambar 2.5 menunjukkan contoh proses convolution yang dijelaskan pada buku (Elgendy, 2020), entri pertama dari feature map (ditandai biru pada Gambar 2.5) dihitung dengan cara menggulung filter bagian bertanda biru pada input image dan feature map dihasilkan dengan mengulangi proses ini untuk setiap elemen input image, output volume dari convolution layer dihasilkan dengan menumpuk feature map setiap filter di sepanjang dimensi kedalaman, setiap komponen activation map dapat dianggap sebagai output dari neuron dan semua neuron dalam activation map juga berbagi parameter satu sama lain.

Menurut (Mostafa & Wu, 2021) karena local connection dari convolutional layer, jaringan dipaksa untuk mempelajari filter yang memiliki respons maksimum

ke wilayah input local. Convolutional layer awal menurut Mustofa menangkap feature tingkat rendah (misalnya, garis) gambar sementara layer selanjutnya mengekstrak feature tingkat tinggi (misalnya, bentuk, objek tertentu). Cara menghitung convolution adalah dengan cara melakukan perkalian dari input image intensitas value gambar dengan filter atau kernel, penulis mencontohkan pada gambar 2.5 dengan kolom yang ditandai biru pada kolom input image yang dilakukan perkalian dengan filter dengan cara $(170 \times 0) + (238 \times -1) + (85 \times 0) + (68 \times 1) + (136 \times 0) + (17 \times 1) + (221 \times 0) + (0 \times 1) + (238 \times 0)$ sehingga mendapatkan value -153 yang diberi tanda biru pada feature map.

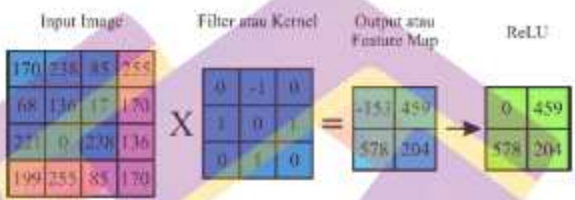


Gambar 2.5 Contoh Proses Convolution

2.3.6. Activation function

Pada buku (Elgendy, 2020) menyatakan ketika membangun sebuah neural network, salah satu keputusan struktur desain yang perlu dibuat adalah activation function apa yang akan digunakan untuk perhitungan neuron yang dibuat. Activation function juga disebut sebagai transfer function atau nonlinier karena mereka mengubah kombinasi linier dari jumlah berbobot menjadi model nonlinier. Ada banyak activation function yang bisa digunakan, namun penulis memilih Rectified Linear (ReLU) untuk arsitektur yang peneliti gunakan, karena berdasarkan penelitian yang dilakukan (Agarap, 2018) ReLU memiliki kelebihan

komputasi yang sederhana dan proses training cepat. ReLU juga merupakan fungsi non-linier dimana pengaktifan neuron tidak dilakukan secara bersamaan dan mengubah value negative menjadi 0, pada gambar 2.6 adalah ilustrasi implementasi ReLU dalam activation function yang pada setiap value negative berubah menjadi value 0.



2.6 Implementasi ReLU Activation

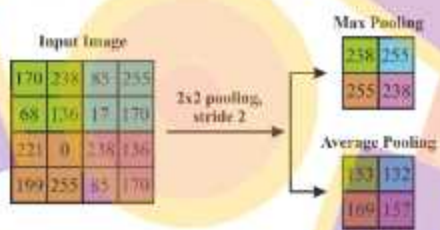
Dalam penggunaannya, fungsi ReLu dituliskan menggunakan persamaan, berikut rumus ReLU yang dijelaskan juga pada penelitian (Firmansyah & Hayadi, 2022):

$$f(x) = \max(0, x) \quad (1)$$

Fungsi mengambil input x dan mengembalikan nilai maksimum antara x dan 0. Dengan kata lain, jika input x negatif, fungsi mengembalikan 0; jika input x positif, fungsi mengembalikan x. Fungsi ini sangat berguna dalam pembelajaran mendalam karena efisien secara komputasi dan memiliki turunan sederhana yaitu 0 atau 1, bergantung pada nilai x. Hal ini memudahkan untuk menghitung gradien selama backpropagation, yang digunakan untuk melakukan training pada neural network.

2.3.7. Pooling Layer

Pooling layer menurut (Yingge et al., 2020) berfungsi untuk mengurangi kompleksitas komputasi, pooling layer dirancang untuk mengurangi dimensi feature map, ini adalah proses diskritisasi sampel. Secara umum menurut Yingge, dua pendekatan digunakan untuk melakukan fungsi ini, seperti yang ditunjukkan pada gambar 2.7. Dijelaskan pada penelitian (Bera & Shrivastava, 2020) bahwa pengumpulan max pooling hasil dari convolution layer sebelumnya sesuai dengan nilai maksimum setiap sub patch dan average pooling adalah nilai hasil dari menghitung rata-rata setiap sub wilayah feature map, dicontohkan pada gambar 2.7 bagian kiri dari input image yang berwarna hijau terdiri dari value 170, 238, 68, 136 maka nilai max pooling adalah 238 dan average pooling adalah 153.

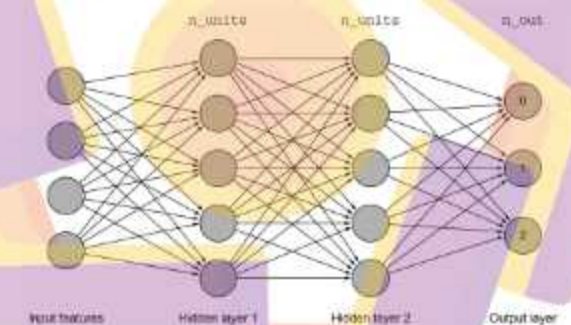


Gambar 2.7 Contoh Pooling Layer

Output dari pooling layer adalah bagian representatif dari data input, bersadarkan dari buku (Elgendy, 2020) operasi pooling diterapkan ke setiap feature map secara terpisah, hal tersebut tidak sensitif dengan perubahan kecil dari layer sebelumnya. Biasanya tidak ada tumpang tindih pada feature map selama prosedur pooling dan langkahnya selalu lebih dari satu untuk pencapaian pengurangan dimensi.

2.3.8. Fully Connected Layer (FCL)

FCL pada intinya menurut (Elgendy, 2020) adalah sebuah neural network multilayer perceptron (MLP), bagian ini adalah proses untuk mendapatkan hasil dari tahap sebelumnya dalam menentukan fitur mana yang paling berkorelasi dengan kelas tertentu dan sering digunakan pada jaringan neural untuk tugas-tugas seperti klasifikasi gambar, pengenalan suara, dan analisis teks. Penelitian yang dilakukan (Wu, 2016) menyatakan penting untuk diketahui bahwa layer sepenuhnya terhubung ke hidden layer berikutnya. Seperti pada gambar 2.8 bahwa setiap node dalam layer terhubung ke semua node di layer sebelumnya yang disebut fully connected layer dan garis pada gambar tersebut adalah bobot yang mewakili pentingnya simpul ini terhadap output value.

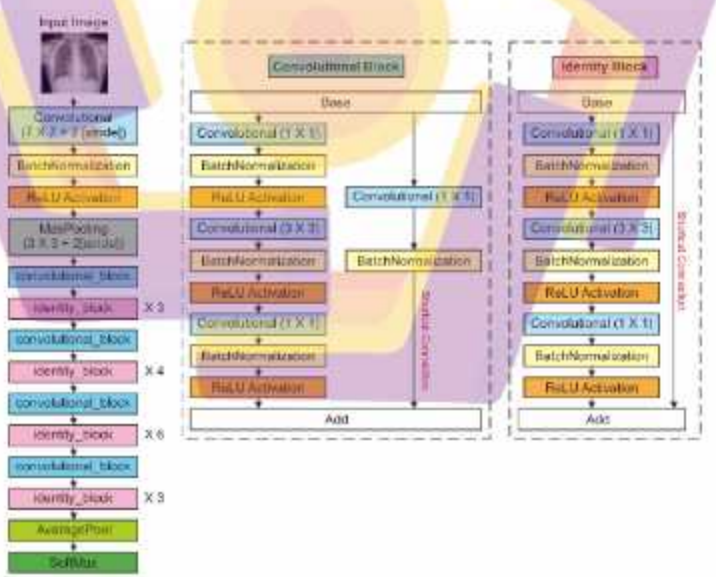


Gambar 2.8 Contoh Terjadinya Fully Connected Layer

Proses FCL pada buku (Chollet, 2017) memiliki setiap neuron yang menerima input dari semua neuron pada layer sebelumnya dan menghitung nilai output dengan menggunakan activation function. Kemudian, nilai output ini menjadi input untuk semua neuron pada layer berikutnya. Oleh karena itu, FCL memiliki banyak parameter yang dapat diatur untuk mengoptimalkan kinerja jaringan neural.

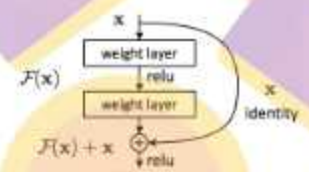
2.3.9. ResNet50

ResNet dijelaskan oleh (Khan et al., 2020) merupakan salah satu arsitektur CNN yang sangat populer dan telah digunakan secara luas pada berbagai tugas pengolahan citra. Arsitektur ini pertama kali diperkenalkan pada tahun 2015 oleh (He et al., 2015) dalam penelitian dengan judul "Deep Residual Learning for Image Recognition" yang dipresentasikan pada konferensi European Conference on Computer Vision (ECCV) 2016. ResNet50 menurut penelitian yang dilakukan (He et al., 2016) adalah versi yang lebih dalam dari arsitektur ResNet yang sebelumnya, dengan 50 lapisan (layer) dan lebih dari 23 juta parameter yang ditunjukkan pada Gambar 2.9.



Gambar 2.9 Alur Proses ResNet50

Dalam penelitian (He et al., 2015) juga menjelaskan dari salah satu keunggulan ResNet50 adalah penggunaan residual block yang memungkinkan jaringan untuk lebih mudah dipelajari dan mengatasi masalah degradasi yang sering terjadi pada CNN dan telah memecahkan masalah ini menggunakan skip connection, hal tersebut adalah connection yang melewati satu atau lebih layer sehingga memberikan jalan memutar untuk melewati gradien tanpa berkurang seperti yang di ilustrasikan pada gambar 2.10, penjelasan *x* merupakan input gambar atau data yang akan di analisa.



Gambar 2.10. Skip Connection Pada Residual Network

ResNet50 dijelaskan oleh (He et al., 2016) memiliki kelemahan utama yaitu 'Vanishing Gradient Problem'. Selama backpropagation, nilai gradien menurun secara signifikan, sehingga hampir tidak ada perubahan pada bobot. Untuk mengatasinya maka ketika digunakan ResNet50 perlu dengan menambahkan "Skip Connection".

2.3.10. MobileNetV2

MobileNetV2 adalah arsitektur CNN yang didesain khusus untuk perangkat mobile dengan sumber daya yang terbatas, seperti smartphone. Arsitektur ini merupakan pengembangan dari arsitektur MobileNet yang sebelumnya, dengan peningkatan kinerja yang signifikan dan pengurangan jumlah parameter.

MobileNetV2 diperkenalkan oleh (Sandler et al., 2018) dalam penelitian berjudul "MobileNetV2: Inverted Residuals and Linear Bottlenecks" yang dipresentasikan pada konferensi IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018.

MobileNetV2 ini secara total mencakup 19 layer, dalam penelitian yang dilakukan oleh (Sandler et al., 2018) layer tengah digunakan untuk membedakan properti dan layer terakhir digunakan untuk klasifikasi. Prosesnya memiliki 3 layer diantara block, tahap pertama adalah layer pertama convolution 1 X 1 dengan ReLU6, layer kedua adalah depthwise convolution, layer ketiga menggunakan convolution 1 X 1 tetapi tanpa ada non-linearity, ini diklaim jika ReLU digunakan lagi maka deep network hanya memiliki power dalam melaktukan klasifikasi linear pada bagian non-zero volume dari output domain seperti pada tabel 2.1 dan keseluruhan arsitektur MobileNetV2 terlihat pada tabel 2.2.

Tabel 2.2. Bottleneck dari MobileNetV2

Input	Operator	Output
$h \times w \times k$	1x1 conv2d, ReLU (6)	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU (6)	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	Linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k^1$

Arsitektur ini dijelaskan juga pada penelitian (Kolonne et al., 2021) bahwa terdiri dari tiga layer, yaitu convolutional layer 1x1 diikuti oleh activation function dari Rectified Linear Unit (ReLU). Layer ini menerima tensor berukuran h x w x k sebagai input, di mana h, w, dan k masing-masing adalah tinggi, lebar, dan jumlah saluran dari peta fitur input. Output dari layer ini adalah tensor dengan ukuran h x w x (t*k), di mana t adalah hyperparameter. Depthwise convolutional layer 3x3

dengan langkah s dan fungsi aktivasi ReLU. Lapisan ini menerima tensor berukuran h x w x (tk) sebagai input dan menghasilkan tensor output berukuran h/s x w/s x (tk). Linear convolutional layer 1x1 yang menggunakan tensor berukuran h/s x w/s x (t*k) sebagai input dan menghasilkan tensor output berukuran h/s x w/s x k.

Tabel 2.3. Arsitektur MobileNetV2

Input	Operator	*t*	*c*	*n*	*s*
224^2 x 3	Conv2d	-	32	1	2
112^2 x 32	Bottleneck	1	16	1	1
112^2 x 16	Bottleneck	6	24	2	2
56^2 x 24	Bottleneck	6	32	3	2
28^2 x 32	Bottleneck	6	64	4	2
14^2 x 64	Bottleneck	6	96	3	1
14^2 x 96	Bottleneck	6	160	3	2
7^2 x 160	bottleneck	6	320	1	1
7^2 x 320	conv2d 1x1	-	1280	1	1
7^2 x 1280	avgpool 7x7	-	-	1	-
1 x 1 x 1280	conv2d 1x1	-	k	-	

Input tensor memiliki ukuran 2242 x 3, dimana 2242 adalah tinggi dan 3 adalah jumlah input channel. Layer pertama adalah convolutional layer dengan 32 filter, ukuran kernel 1x1, langkah 2, dan tanpa padding. Ini mengurangi dimensi spasial tensor input dengan faktor 2 dan menambah jumlah channels menjadi 32. Beberapa layer berikutnya adalah layer "bottleneck". Setiap layer bottleneck terdiri dari tiga sub-layers(Sandler et al., 2018):

- Convolutional layer 1x1 dengan sejumlah filter sama dengan "expansion factor" dikalikan jumlah input channels. Output tensor memiliki dimensi spasial yang sama dengan input tensor.
- Depthwise convolutional layer dengan ukuran kernel 3x3, stride s (yang bervariasi di seluruh layer), dan padding. Ini menerapkan filter terpisah untuk

setiap input channels, menghasilkan output tensor dengan dimensi spasial dan jumlah saluran yang sama dengan input tensor.

- Convolutional layer 1x1 dengan sejumlah filter sama dengan parameter "Output channels". Output tensor memiliki dimensi spasial yang sama dengan input tensor.

Parameter "expansion factor" dan "output channels" bervariasi di seluruh bottleneck layers. Parameter "s" adalah 1 atau 2, tergantung pada layer. Setelah beberapa layer bottleneck, ada final convolutional layer dengan ukuran kernel 1x1, tanpa padding, dan 1280 filter. Ini menghasilkan tensor dengan dimensi spasial 72 x 320 x 1280. Layer berikutnya adalah global average pooling layer dengan ukuran kernel 7x7, yang rata-ratanya melebihi dimensi spasial tensor dan menghasilkan tensor dengan bentuk 1 x 1 x 1280. Layer terakhir adalah convolutional layer 1x1 dengan jumlah filter sama dengan jumlah kelas (k), yang menghasilkan tensor dengan bentuk 1 x 1 x k. Layer ini berfungsi sebagai classifier untuk network.

2.3.11. White Balance

White balance dalam citra medis menurut penelitian yang dilakukan (Siddhartha & Santra, 2020) adalah teknik untuk mengoreksi ketidakseimbangan warna yang terjadi dalam citra. Tujuannya adalah untuk menghilangkan perbedaan warna yang disebabkan oleh variasi pencahayaan atau sumber cahaya saat pengambilan gambar. Ketidakseimbangan warna ini dapat mengganggu visualisasi yang akurat dalam citra medis dan dapat mempengaruhi interpretasi dan analisis citra. White balance dilakukan dengan menyesuaikan intensitas dan distribusi

warna dalam citra sehingga warna yang tampil lebih akurat dan konsisten. Terdapat beberapa metode yang digunakan dalam white balance, termasuk penggunaan target putih referensi atau analisis histogram citra. Penerapan white balance dalam citra medis membantu menghilangkan ketidakseimbangan warna yang tidak diinginkan, meningkatkan visualisasi, dan memastikan keakuratan dalam interpretasi dan analisis citra. Hal ini penting dalam berbagai aplikasi di bidang medis, seperti diagnostik radiologi, analisis citra patologi, penelitian ilmiah, dan pengolahan citra medis lainnya.

2.3.12. Contrast Limited Adaptive Histogram Equalization

CLAHE dijelaskan pada penelitian (Umri et al., 2021) adalah teknik pengolahan citra medis yang digunakan untuk meningkatkan kontras dan detail dalam citra. Teknik ini mengatasi variasi intensitas yang sering terjadi dalam citra medis dengan menerapkan histogram equalization adaptif pada blok-blok kecil citra. Pada penelitian (Roy et al., 2022) juga menyatakan hal ini membantu meningkatkan kontras secara adaptif, dengan mempertimbangkan karakteristik lokal dalam citra. Namun, CLAHE juga memiliki batasan kontras yang terbatas untuk menjaga keseimbangan antara peningkatan kontras dan kualitas citra yang alami. Penerapan CLAHE dalam citra medis membantu memperbaiki visualisasi dan meningkatkan kemampuan analisis citra dalam berbagai aplikasi di bidang medis, seperti segmentasi organ, deteksi tumor, dan analisis struktur tulang.

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis penelitian ini adalah penelitian eksperimental, penelitian ini melakukan pengujian tingkat akurasi yang tertinggi dengan menggunakan Arsitektur ResNet50 dan MobileNetV2 dengan white balance CLAHE menggunakan dua skema dataset. Pengujian ini dilakukan untuk mengetahui metode yang lebih akurat dan tepat dalam melakuan diagnosis penyakit pneumonia pada data CXR.

Penelitian ini bersifat deskriptif, karena menggambarkan suatu objek yang akan diteliti dan menjabarkan hasil pengujian-pengujian yang dilakukan pada dataset yang ada untuk dapat diketahui metode mana yang memiliki accuracy, precision, recall, F1-score, dan time consumtion.

Penelitian ini menggunakan pendekatan kuantitatif yang nantinya hasil dari penelitian ini berupa angka, grafik hasil eksperimen arsitektur ResNet50 dan MobileNetV2.

3.2. Metode Pengumpulan Data

Pengumpulan data dilakukan untuk mendapatkan berbagai macam data yang diperlukan dalam penelitian. Dataset yang digunakan hasil dari website mendeley data yang dikelola oleh Daniel Kermany, Kang Zhang, Michael

Goldbaum bertempat di University of California San Diego dengan link sebagai berikut https://data.mendeley.com/datasets/rscbjbr9sj/2.

3.3. Metode Analisis Data

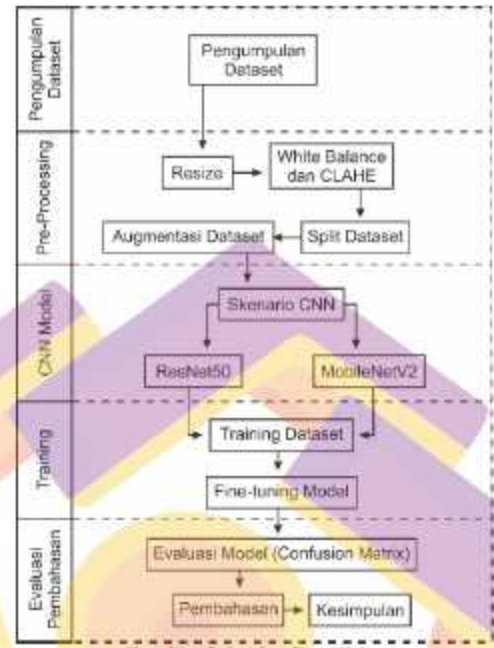
Setelah data-data terkumpul, langkah selanjutnya adalah melakukan analisis untuk mengolah data tersebut menjadi sebuah informasi. Analisis yang dilakukan adalah mendeteksi performa penerapan model ResNet50 dan MobileNetV2 dalam analisis komparasi penyakit pneumonia menggunakan CNN dan melakukan perbandingan dari hasil kedua model tersebut.

3.4. Alur Penelitian

Penelitian memiliki variable eksperiment yang terdiri dari arsitektur, kelas data, epoch, dan parameter terdapat pada tabel 3.1 dan terdapat 5 tahapan inti yang diantaranya yaitu pengumpulan dataset, pre-processing, CNN Model, training, dan evaluasi pembahasan yang didalamnya terdiri dari kumpulan sub proses yang diilustrasikan pada gambar 3.1 dengan rincian sebagai berikut:

Tabel 3.1. Variabel Eksperiment

Arsitektur	Kelas Data	Epoch	Parameter
ResNet-50	2 Kelas	10 Epoch	Accuracy
MobileNetV2	3 Kelas	20 Epoch	Precision
		50 Epoch	Recall
			F1-Score



Gambar 3.1 Alur Penelitian

a. Pengumpulan dataset

Dataset yang digunakan adalah CXR Image dengan dua skema dataset, skema pertama menggunakan 4500 dataset yang terdiri dari tiga kelas, yaitu 1500 CXR normal, 1500 CXR pneumonia bakteri, dan 1500 CXR pneumonia virus. Skema kedua menggunakan 3000 dataset yang terdiri dari dua kelas, yaitu 1500 CXR normal dan 1500 CXR pneumonia.

b. Prepocessing

Pada tahap ini, terdapat beberapa proses yang dilakukan yaitu mulai dari labeling, resize, lalu melakukan splitting data sehingga bisa dilanjutkan kedalam proses selanjutnya yaitu membuat scenario CNN.

i. Resize

Proses resize dilakukan karena dataset yang sudah dikumpulkan tidak memiliki ukuran yang sama dan ukuran citra yang terlalu besar. Resize juga bertujuan untuk menyesuaikan ukuran citra terhadap ukuran data masukan sistem. Pada penelitian ini penulis melakukan resize dengan ukuran 224 X 224.

ii. White Balance dan CLAHE

White balance mengoreksi ketidakseimbangan warna, sementara CLAHE meningkatkan kontras dan detail citra. Kombinasi keduanya meningkatkan visualisasi dan memungkinkan identifikasi pneumonia dengan lebih akurat.

iii. Split data

Terdiri dari dua skema dataset, skema pertama dan skema kedua dilakukan split dataset dengan presentasi pembagian training, validation, dan testing 70:15:15.

iv. Augmentasi Dataset

Augmentasi dataset pada deteksi pneumonia berfungsi untuk mengatasi ketidakseimbangan kelas, meningkatkan variasi data, dan mencegah overfitting model. Dengan variasi baru dari sampel yang ada, augmentasi dataset membantu menciptakan keseimbangan kelas dan meningkatkan kemampuan model dalam mengenali variasi kondisi pneumonia. Selain itu, augmentasi dataset mencegah overfitting dengan memperkenalkan kompleksitas tambahan dalam data latih,

meningkatkan kinerja model dalam mengenali dan membedakan kasus pneumonia pada citra medis.

c. CNN model

i. Skenario CNN

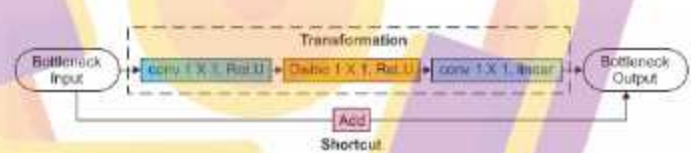
Karena jenis penelitian ini bersifat eksperimen maka alur penelitian ini disediakan langkah skenario. Sebelum peneliti memulai tahap skenario ada beberapa tahap sebelumnya yang dilakukan. Salah satu tahapnya adalah menyiapkan data citra dengan preprocessing sebagai data training dan evaluasi. Untuk dapat mengetahui performa dari peningkatan kualitas citra, maka dilakukan 2 skenario menggunakan algoritma Convolutional Neural Network yaitu ResNet dan MobileNetV2.

ii. ResNet50

Arsitektur ResNet50 menggunakan pembelajaran residual dengan input layer sebagai referensi. Citra masuk ke proses pelatihan pada ResNet50 dengan ukuran 224 x 224 piksel. Citra dikonvolusi dengan filter 7 × 7 dan stride 2, kemudian dinormalisasi oleh Batch Normalization. Dilanjutkan dengan aktivasi menggunakan fungsi ReLU untuk non-linearitas. Setelah itu, dilakukan maxpooling sebelum masuk ke tahap convolution berikutnya. Feature extraction dilakukan melalui kombinasi convolutional block dan identity block. Setelah proses ekstraksi fitur, feature map dijalankan melalui fully connected layer dengan fungsi aktivasi sigmoid untuk prediksi.

iii. MobileNetV2

Arsitektur selanjutnya adalah MobileNetV2 dengan menggunakan depthwise dan pointwise convolution dan ditambah feature linear bottleneck dan shortcut connection diantara bottleneck seperti pada gambar 16. Bagian bottleneck terdapat input dan output antara model sedangkan, layer bagian dalam meng-enkapsulasi kemampuan model untuk mengubah input dari konsep tingkat yang lebih rendah (i.e. piksel) ke deskriptor tingkat yang lebih tinggi (i.e. kategori gambar). Sehingga seperti halnya koneksi residual pada CNN tradisional, shortcut antar bottlenecks memungkinkan training atau pelatihan yang lebih cepat dan akurasi yang lebih baik seperti ilustrasi pada gambar 3.2.



Gambar 3.2 Diagram blok arsitektur

d. Training

Setelah menentukan skenario yang ada yaitu transfer learning menggunakan ResNet50 dan MobileNetV2 kita mulai melakukan percobaan menggunakan dataset gambar X-Ray yang telah dipersiapkan sebelumnya. Setelah itu dilanjutkan dengan tahap fine-tuning untuk meningkatkan performa dan kemampuan generalisasi model yang telah dilatih sebelumnya.

e. Evaluasi dan pembahasan

Tahap ini merupakan tahap untuk melakukan evaluasi (penilaian) berdasarkan tahap pelatihan menggunakan ResNet50 dan MobileNetV2. Ada dua evaluasi yang dilakukan, yaitu mengetahui performa timing penerapan arsitektur ResNet50 dan MobileNetV2 pada tahap peningkatan kualitas citra yang terapkan pada CNN, selanjutnya membandingkan akurasi dari arsitektur ResNet50 dan MobileNetV2 pada peningkatan kualitas citra.

f. Variable penelitian

Pada penelitian

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Pengumpulan Data

Untuk menumpulkan dataset CXR, peneliti mendapatkan dataset dari situs website yang bersifat open data public. Dataset dengan kategori gambar CXR pneumonia (virus dan bakteri) dan CXR normal yang peneliti dapatkan dari satu sumber yaitu Mendeley Data yang berasal dari kontributor (D. Kermany et al., 2018). sumber tersebut sudah menyediakan jenis dataset yang dibutuhkan peneliti dengan jumlah yang besar. Peneliti menggunakan dua skema dataset, skema pertama dengan jumlah 4500 dataset yang dibagi menjadi tiga kelas, terdiri dari 1500 CXR Pneumonia virus, 1500 CXR Pneumonia bakteri, dan 1500 CXR Normal, skema kedua dengan jumlah 3000 dataset yang dibagi menjadi tiga kelas, terdiri dari 1500 CXR Pneumonia dan 1500 CXR Normal. Data tersebut diambil pada tanggal 1 Januari 2023 dan semua gambar memiliki format file jpeg. Informasi lebih lanjut mengenai data penelitian dapat ditemukan pada Tabel 4.1 untuk dataset tiga kelas dan Tabel 4.2 dataset dua kelas.

Tabel 4.1 Dataset CXR Tiga Kelas

NO	Kategori	Jumlah	Gambar
1	Gambar CXR Pneumonia virus	1500	
2	Gambar CXR Pneumonia Bakteri	1500	
3	Gambar CXR Normal	1500	

Tabel 4.2 Dataset CXR Dua Kelas

NO	Kategori	Jumlah	Gambar
1	Gambar CXR Pneumonia	1500	

Tabel 4.2 Dataset CXR Dua Kelas (Lanjutan)

NO	Kategori	Jumlah	Gambar
2	Gambar CXR Normal	1500	

4.2. Preprocessing Data Citra

Setelah berhasil mengumpulkan gambar CXR dengan dua skema dataset yang terdiri dari dua kelas dan tiga kelas. Langkah selanjutnya adalah melakukan pengolahan data untuk meningkatkan kualitas citra dalam proses deteksi dan klasifikasi. Pada tahap ini, akan dijelaskan mengenai proses pre-processing yang akan dilakukan. Dalam tahap ini, akan dijelaskan mengenai proses preprocessing yang akan dilakukan. Berikut adalah langkah-langkah yang akan dilakukan sebagai berikut:

4.2.1. Resize

Tujuan dari proses resize adalah untuk mengubah ukuran gambar yang telah dikumpulkan sehingg sesuai dengan kebutuhan input dari model transfer learning ResNet50 dan MobileNetV2 yang digunakan. Dengan melakukan resize, ukuran gambar akan disesuaikan dengan format yang dibutuhkan oleh sistem. Setelah proses resize selesai, ukuran gambar yang dihasilkan adalah 224 piksel tinggi dan 224 piksel lebar. Hal ini penting untuk memastikan konsistensi ukuran gambar dan

memenuhi persyaratan model yang akan digunakan. Peneliti memberikan logika proses resize sebagai berikut:

```
image = cv2.imread(directory_path + image_name)
image = cv2.resize(image, size)
```

Potongan kode tersebut membuka sebuah gambar menggunakan fungsi `cv2.imread` dan menyimpannya. Kemudian, gambar tersebut diubah ukurannya menggunakan fungsi `cv2.resize` dengan dimensi yang diinginkan yaitu 224 piksel untuk tinggi dan lebar. Gambar yang telah diubah ukurannya kemudian disimpan kembali.

4.2.2. Implementasi White Balance dan CLAHE

White Balance adalah operasi pemrosesan gambar yang digunakan untuk menyesuaikan keakuratan warna dalam gambar digital. Menurut (Siddhartha & Santra, 2020) dalam gambar medis, kondisi pencahayaan rendah dapat menyebabkan beberapa bagian gambar terlihat gelap dan peralatan pengambil gambar tidak mendeteksi cahaya dengan presisi seperti mata manusia. Oleh karena itu, dilakukan pemrosesan atau koreksi gambar untuk memastikan gambar akhir menggambarkan warna yang akurat sesuai dengan gambar aslinya. Tujuan utamanya adalah meningkatkan visibilitas gambar agar dapat mengekstrak fitur yang bermakna. Dengan membuang warna piksel yang jarang muncul dan meregangkan rentang warna yang tersisa, algoritma ini menghindari pengaruh pada batas atas dan bawah gambar yang telah diregangkan. Dalam solusi ini, telah diimplementasikan algoritma white balance menggunakan bahasa Python dengan

menggunakan library NumPy dan OpenCV. Penulis menggunakan algoritma pada tahap white balance sebagai berikut:

$$M_i = P_{0.05}(C)$$

$$M_a = P_{100-0.05}(C)$$

$$C_{upd} = Clip\left(\frac{C - M_i * 255}{M_a - M_i}, 0{,}255\right)$$

Dalam rumus di atas, PiI merupakan representasi pengambilan persentil ke-I dari saluran C, dan operasi Clip (., min, max) menggambarkan operasi saturasi yang dilakukan dengan membatasi nilai dalam rentang min dan max. C dan Cupd melambangkan nilai piksel saluran input dan saluran yang telah diperbarui setelah operasi dilakukan secara berturut-turut. Pada implementasinya menggunakan python sebagai berikut:

```
def white_balance(channel, perc = 0.05):
    mi, ma = (np.percentile(channel, perc), np.percentile(channel,100.0-perc))
    channel = np.uint8(np.clip((channel-mi)*255.0/(ma-mi), 0, 255))
    return channel
```

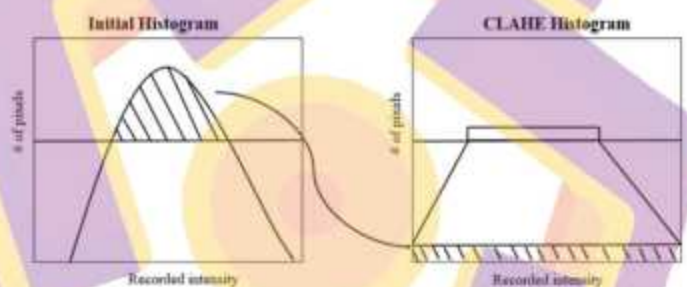
Fungsi tersebut melakukan white balance pada saluran tertentu dari gambar. Ia menghitung nilai minimum (mi) dan maksimum (ma) dengan mengambil persentil saluran tersebut. Kemudian, ia menerapkan algoritma white balancing dengan mengurangkan mi dari saluran, menyesuaikannya ke rentang 0 hingga 255, dan membatasi nilai-nilai agar tetap berada dalam rentang tersebut. Saluran yang telah diimbangi kemudian dikembalikan.

Selanjutnya, proses CLAHE menurut (Umri et al., 2021) histogram dari setiap tile dihitung, yang berkorespondensi dengan bagian-bagian yang berbeda dari gambar, dan digunakan untuk mendapatkan fungsi pemetaan intensitas untuk

setiap tile. Metode ini dapat memperkenalkan noise dalam gambar karena adanya penguatan yang berlebihan. Perhitungan CLAHE dilakukan sebagai berikut:

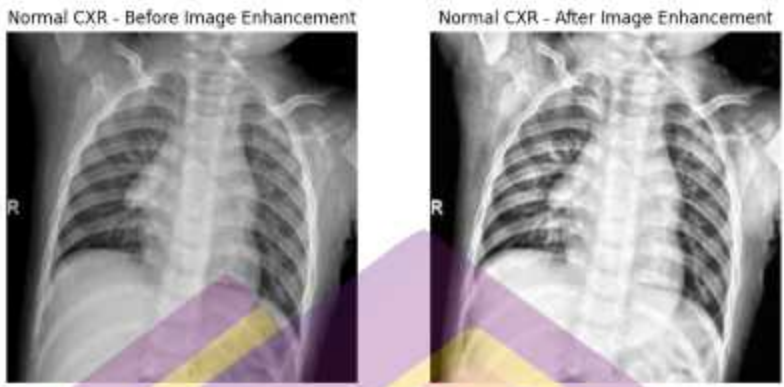
$$P = (P_{max} - P_{min}) * P(f) + P_{min}$$

Dalam rumus di atas, p merepresentasikan nilai piksel setelah menerapkan metode CLAHE, pmax dan pmin merepresentasikan nilai piksel maksimum dan minimum dari gambar secara berturut-turut, dan P(f) merepresentasikan fungsi distribusi probabilitas kumulatif. Jika diilustrasikan fungsi dari CLAHE seperti pada gambar 4.1.



Gambar 4.1 CLAHE

Secara matematis, artinya peneliti menerapkan tren linier ke fungsi distribusi kumulatif. Dalam gambar 4.2 merupakan contoh penerapan CLAHE pada CXR Normal, pada sebelah kiri adalah gambar yang belum masuk tahap CLAHE dan sebelah kanan sudah masuk tahap CLAHE.



Gambar 4.2 Sebelum dan Sesudah CLAHE

Untuk menerapkan CLAHE dalam sistem, dapat dilakukan menggunakan bahasa pemrograman Python dengan menggunakan kode berikut ini:

```
def white_balance(channel, perc = 0.05):
    mi, ma = (np.percentile(channel, perc), np.percentile(channel,100.0-perc))
    channel = np.uint8(np.clip((channel-mi)*255.0/(ma-mi), 0, 255))
    return channel
```

Fungsi tersebut menghasilkan objek CLAHE dengan batasan klip sebesar 2.0 dan ukuran grid tile sebesar (16, 16). Objek CLAHE ini dapat digunakan untuk melakukan equalisasi histogram adaptif dengan batasan klip pada gambar.

4.2.3. Normalize Dataset

Tujuan normalisasi dalam penelitian ini adalah untuk memastikan data berada dalam kondisi yang optimal agar proses pelatihan machine learning dapat dipercepat. Dalam larik citra digital, setiap nilai selalu berada dalam rentang 0 hingga 255. Distorsi yang besar ini dapat memperlambat proses dan pembelajaran model, terutama ketika bobot-bobot kita cenderung berada antara 0 dan 1. Oleh karena itu, dengan mengubah skala model kita melalui pembagian semua nilai

dalam larik dengan 255 (nilai tertinggi yang dapat dicapai dalam citra digital), semua nilai akan berada dalam rentang 0 hingga 1. Hal ini membantu mempercepat pelatihan model dan juga memberikan hasil yang lebih baik.

4.2.4. Split Dataset

Tahap split data adalah proses dimana dataset yang terdapat 2 skenario tersebut dibagi menjadi 3 dataset yang menjadi training data, validasi data, testing data. Berikut contoh script pada python programming yang peneliti lakukan dengan menggunakan dua kelas.

```
image_arrays = [normal_dataset_normalized, pneumonia_dataset_normalized]
datasets = split_and_merge_function(image_arrays, split_factor = [0.7 ,0.15,
0.15])
```

Kode di atas mengacu pada proses pemisahan dataset menggunakan fungsi yang telah ditentukan sebelumnya. Terdapat dua array, yaitu `normal_dataset_normalized` yang berisi dataset gambar normal yang telah dinormalisasi dan `pneumonia_dataset_normalized` yang berisi dataset gambar pneumonia yang telah dinormalisasi. Kedua array tersebut digabungkan menjadi satu dalam array image_arrays.

Selanjutnya, dilakukan pemisahan dataset menggunakan fungsi `split_and_merge_function`. Fungsi ini menerima input berupa array gambar dan faktor pemisahan yang menentukan proporsi pemisahan dataset menjadi training set, validation set, dan test set. Dalam contoh ini, faktor pemisahan yang digunakan adalah [0.7, 0.15, 0.15], yang berarti 70% dataset akan menjadi training set, 15% akan menjadi validation set, dan 15% akan menjadi test set.

Hasil pemisahan dataset disimpan dalam variabel `datasets`. Variabel ini akan berisi dataset yang telah terpisah sesuai dengan proporsi yang ditentukan, siap digunakan dalam proses training, validasi, dan pengujian model.

4.2.5. Augmentasi Data

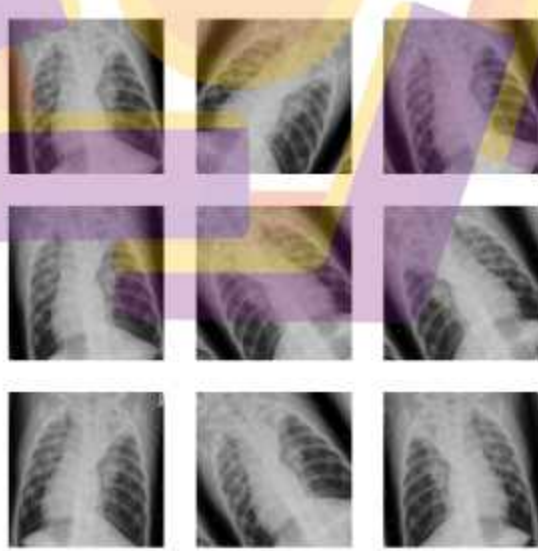
Overfitting terjadi karena jumlah sampel yang terlalu sedikit untuk dipelajari, sehingga membuat kita tidak dapat melatih model yang mampu melakukan generalisasi pada data baru. Jika data yang tersedia tak terbatas, model kita akan terpapar pada setiap aspek yang mungkin dari distribusi data yang ada: kita tidak akan mengalami overfitting. Augmentasi data mengambil pendekatan untuk menghasilkan lebih banyak data pelatihan dari sampel pelatihan yang ada, dengan melakukan "pengaugmentasian" sampel-sampel tersebut melalui sejumlah transformasi acak yang menghasilkan gambar-gambar yang terlihat meyakinkan. Tujuan dari hal ini adalah agar pada saat pelatihan, model kita tidak pernah melihat gambar yang persis sama dua kali. Pendekatan ini membantu model untuk terpapar pada lebih banyak aspek data dan meningkatkan kemampuannya dalam melakukan generalisasi dengan melakukan transiasi, transformasi, penambahan noise, rotasi, dan pembesaran.

Pada penelitian ini, awalnya terdapat 4500 data dengan 3 kategori yang berbeda. Setelah mengalami proses augmentasi data, jumlah data yang tersedia menjadi 9000 dengan 3 kategori yang berbeda. Selanjutnya, terdapat 3000 data dengan 2 kategori yang berbeda setelah proses augmentasi data, sehingga total data

yang tersedia menjadi 6000 dengan 2 kategori yang berbeda. Berikut adalah contoh perintah yang akan dieksekusi dalam program seperti di bawah ini.

```
def data_augmenter():
    data_augmentation = tf.keras.Sequential()
    data_augmentation.add(RandomFlip('horizontal'))
    data_augmentation.add(RandomRotation(0.1))
    return data_augmentation
```

Fungsi `data_augmenter()` ini adalah untuk membuat sebuah objek model sequential dalam TensorFlow dengan menggunakan dua operasi augmentasi data, yaitu `RandomFlip` dan `RandomRotation`. Operasi `RandomFlip` digunakan untuk secara acak membalikkan gambar secara horizontal, sedangkan operasi `RandomRotation` digunakan untuk secara acak memutar gambar sebesar 0.1 radian. Setelah menambahkan kedua operasi augmentasi, objek model sequential tersebut akan dikembalikan sebagai output. Selanjutnya, gambar-gambar hasil augmentasi dapat dilihat pada Gambar 4.3.



Gambar 4.3 Menampilkan Hasil dari Augmentasi Data Citra X-Ray

4.3. Skenario Model CNN

Dalam tahap persiapan skenario, peneliti akan membuat skenario eksperimental sesuai dengan jenis penelitian yang bersifat eksperimental. Pada penelitian ini, peneliti akan mencoba bereksperimen dengan menggunakan algoritma dan metode yang telah ditetapkan sebelumnya. Arsitektur yang akan digunakan dalam skenario ini adalah ResNet50 dan MobileNetV2, yang merupakan arsitektur layer yang sudah ada sebelumnya. Penelitian sebelumnya telah menunjukkan hasil yang optimal dalam penerapan transfer learning menggunakan model ResNet50 dan MobileNetV2. Oleh karena itu, penulis tertarik untuk menguji kedua model tersebut pada dataset yang diajukan dan menggabungkannya dengan teknik peningkatan kualitas citra. Tujuan penelitian ini adalah untuk mencapai hasil yang lebih baik dalam mengklasifikasikan gambar pada dataset dengan memanfaatkan kekuatan transfer learning dan peningkatan kualitas citra.

Penelitian (Kumar Sethy et al., 2020) menekankan pentingnya deteksi COVID-19 menggunakan citra X-Ray. Metode deep learning dengan ResNet-50 dan SVM berhasil mencapai akurasi, F1 score, dan metrik evaluasi lainnya sebesar 95,38%, 91,41%, dan 90,76% dalam mengidentifikasi pasien terinfeksi. Model klasifikasi ini unggul dibandingkan dengan metode lain dan bermanfaat bagi praktisi medis dalam diagnosis COVID-19. Penelitian lainnya yang dilakukan (Roy et al., 2022), mengusulkan teknik augmentasi data baru bernama SVD-CLAHE Boosting dan loss function baru bernama BWCCE untuk mendeteksi penyakit Covid-19 dari gambar CXR. Metode Boosting SVD-CLAHE yang digunakan melibatkan oversampling dan under-sampling. Hasil eksperimen menunjukkan

bahwa model ResNet-50 yang dilatih dengan data yang diperbesar menggunakan SVD-CLAHE Boosting dan BWCCE loss function mencapai hasil yang lebih baik dibandingkan model CNN konvensional lainnya dan model Covid-Lite dan Covid-Net. F1 score mencapai 95%, accuracy 94%, recall 96%, dan precision 96%.

Alasan selanjutnya berdasarkan dari Penelitian oleh (Dilshad et al., 2021)mengusulkan penggunaan model Convolutional Neural Network (CNN) berbasis MobileNetV2 untuk deteksi dini Covid-19 dari gambar CXR. Tujuan penelitian ini adalah memberikan skrining awal dengan daya komputasi rendah untuk membantu ahli radiologi dalam identifikasi pasien potensial Covid-19. Pengujian dilakukan menggunakan dataset 894 gambar CXR yang dibagi menjadi empat bagian. Hasil eksperimen menunjukkan akurasi 96,33%, F1-score 93-96%, dan sensitivitas 93-96%. Tingkat false negative sangat rendah, bahkan mencapai 0% pada dataset lokal. Waktu eksekusi model sangat cepat, kurang dari 0,1 detik.

Terdapat 2 skenario dataset dan 3 skenario epoch yang akan digunakan untuk kedua model ResNet50 dan MobileNetV2 pada penelitian ini. Selain itu, akan ditambahkan teknik pemrosesan citra seperti white balance dan CLAHE sebagai metode peningkatan kualitas citra. Hal ini bertujuan untuk mengevaluasi pengaruh penggunaan peningkatan kualitas citra terhadap akurasi sesuai dengan rumusan masalah penelitian. Berikut adalah susunan skenario yang akan digunakan:

4.3.1. ResNet50

arsitektur jaringan saraf konvolusi (CNN) yang populer dalam bidang pengolahan citra. Arsitektur ini dikembangkan oleh Kaiming He dan timnya pada

tahun 2015 (He et al., 2015). ResNet50 terdiri dari 50 lapisan (hence the name "50") dan menggunakan blok residual untuk memperkuat pembelajaran dalam jaringan yang lebih dalam. ResNet50 telah terbukti sangat efektif dalam berbagai tugas pengolahan citra, seperti klasifikasi gambar, deteksi objek, dan segmentasi. Dengan memiliki 50 lapisan, arsitektur ini dapat menangani dan mempelajari fitur-fitur kompleks dalam citra dengan lebih baik, sehingga menghasilkan representasi yang lebih akurat dan pemahaman yang lebih dalam terhadap data citra.

Dalam skenario pertama menggunakan arsitektur ResNet50, tahap awal yang dilakukan adalah preprocessing data. Preprocessing ini melibatkan pengolahan dan persiapan data agar siap digunakan sebagai input pada tahap pembelajaran.

```
def make_resnet_model(image_size, num_classes,
data_augmentation=data_augmenter()):
    input_shape = image_size + (3,)

    base_model = tf.keras.applications.ResNet50(
        include_top=False,
        weights="imagenet",
        input_shape=input_shape,
    )
    base_model.trainable = False
    inputs = tf.keras.Input(shape=input_shape)
    x = data_augmentation(inputs)
    x = base_model(x, training=False)
    x = GlobalAveragePooling2D()(x)
    x = Dropout(0.2)(x)
    if num_classes == 2:
        activation = "sigmoid"
        units = 2
    else:
        activation = "softmax"
        units = num_classes
    x = Dropout(0.5)(x)
    prediction_layer = Dense(units, activation=activation)
    outputs = prediction_layer(x)
    model = Model(inputs, outputs)

    return model
```

Fungsi make_resnet_model membuat model ResNet50 untuk klasifikasi citra. Fungsi ini menerima parameter image_size (ukuran citra masukan), num_classes (jumlah kelas untuk klasifikasi), dan fungsi opsional data_augmentation (digunakan untuk augmentasi data). Pertama, bentuk input (input shape) ditentukan berdasarkan ukuran citra. Model ResNet50 kemudian dibuat dengan menggunakan bobot pra-pelatihan dari dataset ImageNet dan bentuk input yang telah ditentukan sebelumnya. Model dasar (base model) diatur agar tidak dapat dilatih (non-trainable).

Selanjutnya, input layer (input layer) dibuat dan augmentasi data diterapkan pada masukan menggunakan fungsi data_augmentation. Model dasar kemudian diterapkan pada masukan yang telah di-augmentasi dengan pengaturan training=False untuk menghindari pembaruan statistik normalisasi batch.

layer tambahan ditambahkan untuk klasifikasi, termasuk layer pooling rata-rata global (global average pooling), lapisan dropout, dan lapisan dense dengan jumlah unit dan fungsi aktivasi yang sesuai berdasarkan jumlah kelas. Terakhir, model dibuat dengan menggunakan input dan output yang telah ditentukan, dan model tersebut dikembalikan. Selanjutnya pada peneliti menampilkan susunan layer dan membagi klasifikasi yang pada program sebagai berikut.

```
image_size = (224, 224)
resnet_model = make_resnet_model(image_size, num_classes=2)

resnet_model.summary()
```

Fungsi program di atas membuat model ResNet50 dengan menggunakan ukuran citra (image_size) 224x224 dan 2 kelas untuk klasifikasi. Kemudian, fungsi `make_resnet_model` dipanggil dengan parameter yang sesuai. Setelah model

dibuat, fungsi `summary()` digunakan untuk menampilkan ringkasan (summary) dari model ResNet50 yang telah dibuat yang dapat diliat pada tabel 4.3.

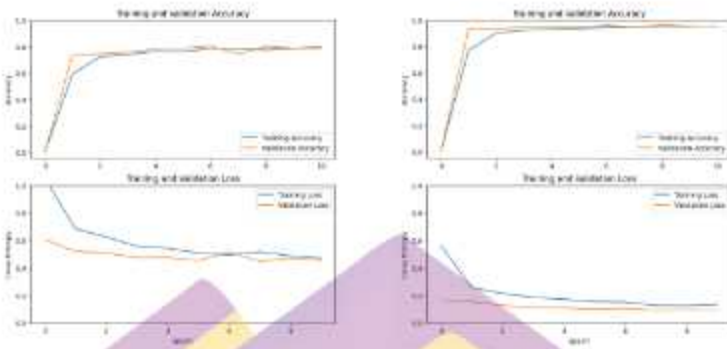
Table 4.3 Layer Model ResNet50

Layer (type)	**Output Shape**	**Param #**
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
sequential_1 (Sequential)	(None, 224, 224, 3)	0
resnet50 (Functional)	(None, 7, 7, 2048)	23587712
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dropout (Dropout)	(None, 2048)	0
dropout_1 (Dropout)	(None, 2048)	0
dense (Dense)	(None, 2)	4098
Total params: 23,591,810		
Trainable params: 4,098		
Non-trainable params: 23,587,712		

Training validation data menggunakan 2 skenario yaitu dengan 3 kelas (Normal CXR, Bakteri Pneumonia CXR, dan Virus Pneumonia CXR) dan dua kelas (Normal CXR dan Pneumonia CXR).

4.3.1.1. Training Validation 10 Epoch

Berikut adalah hasil akurasi dan loss pada data training dan data validasi dalam gambar 4.4 untuk (a) 3 kelas dan (b) 2 kelas.



(a) (b)

Gambar 4.4 ResNet50 10 Epoch. (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.4 dan tabel 4.4, memperlihatkan bahwa tahap training 3 kelas dengan training awal (Epoch 1), akurasi training model sebesar 59,27% dengan validation accuracy sebesar 73,48%. Seiring dengan peningkatan epoch, akurasi model mengalami peningkatan secara bertahap hingga mencapai 79,17% pada Epoch 10 dengan validation accuracy sebesar 80%. Selain itu, loss mengalami penurunan pada setiap epoch, menunjukkan bahwa model semakin baik dalam meminimalkan kesalahan.

Tabel 4.4 Hasil ResNet50 10 Epoch 3 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/10	75ms	1.0541	0.5927	0.6062	0.7348	0.0010
2/10	36ms	0.6843	0.7248	0.5213	0.7481	0.0010
3/10	35ms	0.6262	0.7416	0.5095	0.7585	0.0010
4/10	36ms	0.5563	0.7654	0.4744	0.7837	0.0010
5/10	35ms	0.5439	0.7625	0.4761	0.7896	0.0010
6/10	35ms	0.5075	0.7867	0.4544	0.8059	0.0010
7/10	33ms	0.4953	0.7829	0.5162	0.7452	0.0010
8/10	34ms	0.5172	0.7762	0.4493	0.8030	0.0010
9/10	33ms	0.4875	0.7867	0.4716	0.7896	0.0010
10/10	33ms	0.4704	0.7917	0.4549	0.8000	0.0010

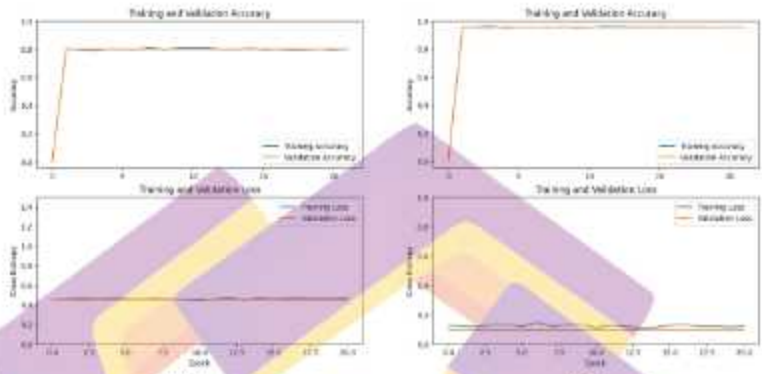
Sedangkan untuk 2 kelas, pada awal pelatihan (Epoch 1), akurasi model sebesar 77,29% dengan validation accuracy sebesar 93,56%. Selama pelatihan, akurasi model terus meningkat secara bertahap hingga mencapai 95,76% pada Epoch 10, dengan validation accuracy sebesar 95,33%. Training loss juga mengalami penurunan yang signifikan dari 56,54% pada Epoch 1 menjadi 13,46% pada Epoch 10 dan validation loss dari 16,84% menjadi 9,52% yang dapat dilihat pada tabel 4.5.

Tabel 4.5 Hasil ResNet50 10 Epoch 2 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/10	101ms	0.5654	0.7729	0.1684	0.9356	0.0010
2/10	34ms	0.2569	0.9029	0.1560	0.9333	0.0010
3/10	37ms	0.2146	0.9252	0.1277	0.9511	0.0010
4/10	33ms	0.1856	0.9319	0.1144	0.9489	0.0010
5/10	33ms	0.1722	0.9386	0.1110	0.9467	0.0010
6/10	37ms	0.1550	0.9457	0.0994	0.9644	0.0010
7/10	34ms	0.1519	0.9486	0.1066	0.9467	0.0010
8/10	37ms	0.1274	0.9548	0.0931	0.9667	0.0010
9/10	33ms	0.1256	0.9519	0.0966	0.9556	0.0010
10/10	33ms	0.1346	0.9576	0.0952	0.9533	0.0010

Dari hasil tersebut dapat disimpulkan bahwa model dengan 2 kelas 3 kelas yang ditraining dengan 10 epoch sudah mampu mencapai akurasi yang relatif tinggi. Akurasi yang tinggi dan penurunan loss menunjukkan bahwa model mampu mempelajari pola-pola yang relevan dalam data training dan dapat melakukan prediksi yang lebih akurat pada data validasi. Hal ini menunjukkan kemampuan model untuk menggeneralisasi dan beradaptasi dengan baik pada data yang belum pernah dilihat sebelumnya. Setelah itu, peneliti melakukan perbaikan menggunakan fine tuning yang berfungsi untuk melakukan penyesuaian learning rate yang diatur

dari mulai terkecil sampai terbesar dengan menambahkan 30 epoch, dengan hasil pada gambar 4.5.



(a) (b)

Gambar 4.5 Fine Tuning ResNet50 10 Epoch. (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Berdasarkan gambar pada 4.5 dan tabel 4.6 fine tuning yang dilakukan pada 3 kelas, bahwa model yang digunakan tidak mengalami peningkatan performa yang signifikan selama proses pelatihan. Model awalnya mencapai akurasi training sebesar 80,22% dengan validation accuracy sebesar 80,00% pada Epoch 10. Selanjutnya, pada Epoch 18 dilakukan penurunan learning rate menggunakan metode ReduceLROnPlateau, namun hal ini tidak membawa perubahan yang signifikan dalam performa model.

Tabel 4.6 Hasil Fine tuning ResNet50 10 Epoch 3 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
10/40	60ms	0.4539	0.8022	0.4549	0.8000	1.0000e-04
11/40	32ms	0.4583	0.7946	0.4549	0.8000	1.0000e-04
12/40	32ms	0.4698	0.7873	0.4549	0.8000	1.0000e-04
13/40	32ms	0.4613	0.8010	0.4549	0.8000	1.0000e-04
14/40	32ms	0.4666	0.7981	0.4549	0.8000	1.0000e-04
15/40	33ms	0.4645	0.7968	0.4549	0.8000	1.0000e-04

Tabel 4.6 Hasil Fine tuning ResNet50 10 Epoch 3 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
16/40	33ms	0.4574	0.8105	0.4549	0.8000	1.0000e-04
17/40	33ms	0.4672	0.7962	0.4549	0.8000	1.0000e-04
18/40	33ms	0.4588	0.8083	0.4549	0.8000	1.0000e-04
19/40	32ms	0.4531	0.8095	0.4549	0.8000	1.0000e-05
20/40	32ms	0.4472	0.8089	0.4549	0.8000	1.0000e-05
21/40	33ms	0.4594	0.7984	0.4549	0.8000	1.0000e-05
22/40	32ms	0.4769	0.7971	0.4549	0.8000	1.0000e-05
23/40	33ms	0.4493	0.8048	0.4549	0.8000	1.0000e-05
24/40	32ms	0.4756	0.7949	0.4549	0.8000	1.0000e-05
25/40	33ms	0.4592	0.7990	0.4549	0.8000	1.0000e-05
26/40	33ms	0.4698	0.7924	0.4549	0.8000	1.0000e-05
27/40	32ms	0.4704	0.7949	0.4549	0.8000	2.0000e-06
28/40	33ms	0.4629	0.7997	0.4549	0.8000	2.0000e-06
29/40	32ms	0.4676	0.7921	0.4549	0.8000	2.0000e-06
30/40	36ms	0.4721	0.8029	0.4549	0.8000	2.0000e-06

Sedangkan hasil finetuning pada 2 kelas, epoch awal model mencapai akurasi training sebesar 95,05% dan validation accuracy sebesar 95,33%. Namun, selama 10 epoch berikutnya, baik akurasi training maupun validation accuracy tetap pada angka tersebut, yaitu sekitar 95,00%. Meskipun ada percobaan untuk menurunkan learning rate pada epoch 18 menggunakan metode ReduceLROnPlateau, hal ini tidak membawa perubahan yang signifikan dalam performa model. Kesimpulannya, model tampaknya sudah mencapai batasannya dalam mempelajari pola dari data pelatihan yang dapat dilihat pada tabel 4.7.

Tabel 4.7 Hasil Fine tuning ResNet50 10 Epoch 2 Kelas

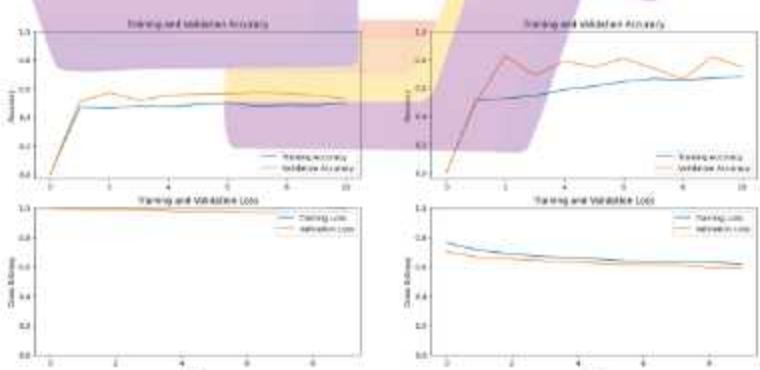
Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
10/40	75ms	0.1261	0.9505	0.0952	0.9533	1.0000e-04
11/40	32ms	0.1212	0.9567	0.0952	0.9533	1.0000e-04
12/40	33ms	0.1167	0.9610	0.0952	0.9533	1.0000e-04
13/40	33ms	0.1320	0.9495	0.0952	0.9533	1.0000e-04

Tabel 4.7 Hasil Fine tuning ResNet50 10 Epoch 2 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
14/40	33ms	0.1313	0.9533	0.0952	0.9533	1.0000e-04
15/40	32ms	0.1188	0.9557	0.0952	0.9533	1.0000e-04
16/40	33ms	0.1459	0.9514	0.0952	0.9533	1.0000e-04
17/40	33ms	0.1183	0.9571	0.0952	0.9533	1.0000e-04
18/40	33ms	0.1324	0.9510	0.0952	0.9533	1.0000e-04
19/40	32ms	0.1309	0.9519	0.0952	0.9533	1.0000e-05
20/40	33ms	0.1117	0.9595	0.0952	0.9533	1.0000e-05
21/40	33ms	0.1269	0.9590	0.0952	0.9533	1.0000e-05
22/40	33ms	0.1192	0.9571	0.0952	0.9533	1.0000e-05
23/40	33ms	0.1126	0.9562	0.0952	0.9533	1.0000e-05
24/40	33ms	0.1139	0.9581	0.0952	0.9533	1.0000e-05
25/40	33ms	0.1272	0.9552	0.0952	0.9533	1.0000e-05
26/40	33ms	0.1330	0.9567	0.0952	0.9533	1.0000e-05
27/40	33ms	0.1185	0.9571	0.0952	0.9533	2.0000e-06
28/40	32ms	0.1200	0.9533	0.0952	0.9533	2.0000e-06
29/40	32ms	0.1158	0.9562	0.0952	0.9533	2.0000e-06
30/40	37ms	0.1195	0.9533	0.0952	0.9533	2.0000e-06

4.3.1.2. Training Validation 10 Epoch (CLAHE – White Balance)

Berikut adalah hasil akurasi dan loss pada data training dan data validasi dalam gambar 4.6 untuk (a) 3 kelas dan (b) 2 kelas.



(a) (b)

Gambar 4.6 ResNet50 10 Epoch (CLAHE – White Balance). (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.6 dan tabel 4.8, dapat dilihat hasil pada 3 kelas pada awal loss training adalah 102,42% dan akurasi training adalah 47,02%, sedangkan loss validation adalah 99,52% dan akurasi validation adalah 50,81%. Selama training, terjadi fluktuasi dalam akurasi dan loss pada data training maupun validation. Akurasi training berkisar antara 46,29% dan 50,03%, sedangkan akurasi validation berkisar antara 51,85% dan 57,19%. Pada epoch ke-10, learning rate mengalami penurunan, sehingga dapat membantu meningkatkan kestabilan dan keakuratan model.

Tabel 4.8 Hasil ResNet50 10 Epoch 3 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/10	62ms	1.0242	0.4702	0.9952	0.5081	0.0010
2/10	50ms	1.0226	0.4629	0.9894	0.5719	0.0010
3/10	47ms	1.0119	0.4784	0.9903	0.5185	0.0010
4/10	47ms	1.0166	0.4749	0.9876	0.5541	0.0010
5/10	47ms	1.0097	0.4889	0.9740	0.5600	0.0010
6/10	47ms	1.0001	0.5003	0.9779	0.5630	0.0010
7/10	47ms	1.0000	0.4787	0.9670	0.5719	0.0010
8/10	47ms	0.9963	0.4863	0.9659	0.5659	0.0010
9/10	47ms	1.0016	0.4835	0.9720	0.5526	0.0010
10/10	47ms	0.9922	0.4994	0.9635	0.5304	0.0010

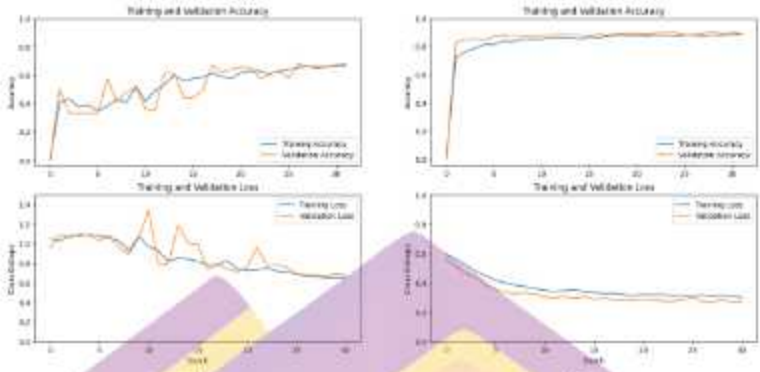
Selanjutnya training pada 2 kelas, awalnya loss training adalah 76,20% dan akurasi adalah 51%, sedangkan loss validation adalah 70,01% dan akurasi validation adalah 50%. Namun selama training terjadi peningkatan yang signifikan dalam akurasi dan penurunan dalam loss. Pada epoch kedua, akurasi training meningkat menjadi 52,62% dan akurasi validation meningkat menjadi 82,89%. Pada epoch selanjutnya, akurasi training terus meningkat secara bertahap dan

mencapai 68% pada epoch ke-10. Sementara itu, akurasi validation naik dan turun, mencapai tingkat tertinggi pada epoch ke-3 dengan 82,22%. Selama pelatihan, learning rate juga mengalami penurunan pada epoch ke-10. Hal ini dapat membantu meningkatkan kestabilan dan keakuratan model yang dapat dilihat pada tabel 4.9.

Tabel 4.9 Hasil ResNet50 10 Epoch 2 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/10	122ms	0.7620	0.5100	0.7001	0.5000	0.0010
2/10	52ms	0.7113	0.5262	0.6636	0.8289	0.0010
3/10	49ms	0.6854	0.5443	0.6523	0.6889	0.0010
4/10	48ms	0.6684	0.5881	0.6393	0.7911	0.0010
5/10	49ms	0.6588	0.6129	0.6301	0.7467	0.0010
6/10	49ms	0.6451	0.6448	0.6184	0.8111	0.0010
7/10	49ms	0.6318	0.6657	0.6137	0.7400	0.0010
8/10	48ms	0.6330	0.6567	0.6138	0.6622	0.0010
9/10	49ms	0.6302	0.6690	0.5946	0.8222	0.0010
10/10	49ms	0.6186	0.6800	0.5942	0.7489	0.0010

Kesimpulannya, model ResNet50 menunjukkan peningkatan dalam akurasi dan penurunan dalam loss seiring dengan berjalannya epoch. Namun, ada fluktuasi dalam akurasi validasi, yang dapat menunjukkan kemungkinan terjadinya overfitting. Untuk melakukan peningkatan hasil maka peneliti melakukan perbaikan menggunakan fine tuning yang berfungsi untuk melakukan penyesuaian learning rate yang diatur dari mulai terkecil sampai terbesar dengan menambahkan 30 epoch, dengan hasil pada gambar 4.7.



(a) (b)

Gambar 4.7 Fine Tuning ResNet50 10 Epoch (CLAHE – White Balance). (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.7 dan tabel 4.10 dapat dilihat setelah melalui finetuning pada 3 kelas, peneliti melihat bahwa akurasi training sebesar 66,51% dan akurasi validation sebesar 68%. Selama training, peneliti mengurangi learning rate jika tidak ada peningkatan dalam akurasi validation selama beberapa epoch berturut-turut. Dalam grafik training, dapat melihat bahwa seiring dengan peningkatan jumlah epoch, loss berkurang baik pada data training maupun data validation. Namun, akurasi tampaknya stagnan pada tingkat sekitar 66%-68%.

Tabel 4.10 Hasil Fine tuning ResNet50 10 Epoch 3 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
10/40	67ms	1.0517	0.4092	0.9552	0.5052	1.0000e-04
11/40	62ms	1.0275	0.4349	1.0966	0.3333	1.0000e-04
12/40	62ms	1.0878	0.3778	1.0545	0.3333	1.0000e-04
13/40	63ms	1.0787	0.3914	1.1007	0.3333	1.0000e-04
14/40	62ms	1.0941	0.3476	1.0813	0.3333	1.0000e-04
15/40	64ms	1.0790	0.3838	1.0324	0.5733	1.0000e-04
16/40	61ms	1.0572	0.4298	1.0920	0.4133	1.0000e-04
17/40	61ms	1.0249	0.4076	0.9551	0.4815	1.0000e-04
18/40	62ms	0.9281	0.5238	0.8952	0.5067	1.0000e-04

Tabel 4.10 Hasil Fine tuning ResNet50 10 Epoch 3 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
19/40	62ms	1.0746	0.4162	1.0630	0.3644	1.0000e-04
20/40	61ms	0.9713	0.4914	1.3391	0.3556	1.0000e-04
21/40	63ms	0.9250	0.5346	0.7798	0.6178	1.0000e-04
22/40	61ms	0.8209	0.5987	0.8053	0.6119	1.0000e-04
23/40	61ms	0.8547	0.5616	1.1838	0.4430	1.0000e-04
24/40	61ms	0.8425	0.5765	1.0042	0.4415	1.0000e-04
25/40	61ms	0.8225	0.5854	0.9922	0.4963	1.0000e-04
26/40	69ms	0.7754	0.6127	0.7372	0.6696	1.0000e-04
27/40	61ms	0.7912	0.5914	0.7935	0.6222	1.0000e-04
28/40	61ms	0.8203	0.5752	0.7402	0.6415	1.0000e-04
29/40	62ms	0.7439	0.6210	0.7154	0.6578	1.0000e-04
30/40	61ms	0.7247	0.6292	0.7594	0.6548	1.0000e-04
31/40	61ms	0.7307	0.6292	0.9613	0.5778	1.0000e-04
32/40	63ms	0.7536	0.6095	0.7793	0.6030	1.0000e-04
33/40	61ms	0.7202	0.6251	0.7845	0.6341	1.0000e-04
34/40	61ms	0.7118	0.6390	0.7563	0.5763	1.0000e-04
35/40	64ms	0.6896	0.6543	0.6921	0.6830	1.0000e-05
36/40	61ms	0.6699	0.6629	0.6797	0.6607	1.0000e-05
37/40	61ms	0.6644	0.6514	0.6817	0.6622	1.0000e-05
38/40	61ms	0.6509	0.6600	0.6691	0.6607	1.0000e-05
39/40	61ms	0.6525	0.6648	0.6923	0.6741	1.0000e-05
40/40	61ms	0.6512	0.6651	0.6776	0.6800	1.0000e-05

Sedangkan pada gambar tersebut dapat dilihat hasil fine tuning 2 kelas, terlihat bahwa model mencapai tingkat akurasi yang cukup baik di atas 85% pada dataset validasi. Namun, melihat pada grafik loss dan akurasi bahwa model mengalami peningkatan dalam kinerja pada awal training, tetapi kemudian kinerja tersebut mulai stagnan yang dapat dilihat pada tabel 4.11.

Tabel 4.11 Hasil Fine tuning ResNet50 10 Epoch 2 Kelas (CLAHE – White Balance)

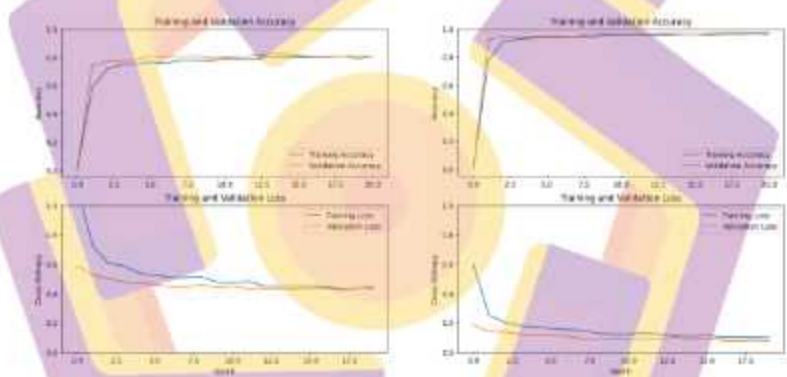
Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
10/40	82ms	0.5929	0.7229	0.5575	0.8311	2.0000e-06
11/40	67ms	0.5614	0.7648	0.5117	0.8511	2.0000e-06
12/40	67ms	0.5213	0.7876	0.4653	0.8533	2.0000e-06
13/40	63ms	0.4763	0.8148	0.4397	0.8467	2.0000e-06
14/40	66ms	0.4484	0.8124	0.3835	0.8733	2.0000e-06
15/40	63ms	0.4165	0.8400	0.3540	0.8733	2.0000e-06
16/40	66ms	0.3993	0.8357	0.3357	0.8800	2.0000e-06
17/40	63ms	0.3839	0.8529	0.3260	0.8778	2.0000e-06
18/40	63ms	0.3744	0.8490	0.3328	0.8689	2.0000e-06
19/40	63ms	0.3607	0.8519	0.3239	0.8733	2.0000e-06
20/40	67ms	0.3502	0.8652	0.3043	0.8844	2.0000e-06
21/40	67ms	0.3406	0.8610	0.2983	0.8867	2.0000e-06
22/40	63ms	0.3485	0.8610	0.3066	0.8844	2.0000e-06
23/40	63ms	0.3544	0.8543	0.2961	0.8822	2.0000e-06
24/40	63ms	0.3393	0.8690	0.3072	0.8778	2.0000e-06
25/40	63ms	0.3331	0.8643	0.2889	0.8867	2.0000e-06
26/40	64ms	0.3281	0.8752	0.2976	0.8822	2.0000e-06
27/40	67ms	0.3288	0.8805	0.2838	0.8911	2.0000e-06
28/40	67ms	0.3206	0.8748	0.2840	0.8933	2.0000e-06
29/40	63ms	0.3143	0.8781	0.2816	0.8933	2.0000e-06
30/40	63ms	0.3214	0.8738	0.2874	0.8889	2.0000e-06
31/40	67ms	0.3203	0.8786	0.2823	0.8956	2.0000e-06
32/40	67ms	0.3208	0.8781	0.2766	0.8978	2.0000e-06
33/40	63ms	0.3111	0.8743	0.2748	0.8978	2.0000e-06
34/40	63ms	0.3130	0.8795	0.2847	0.8844	2.0000e-06
35/40	62ms	0.3139	0.8805	0.3031	0.8844	2.0000e-06
36/40	63ms	0.3178	0.8724	0.2715	0.8956	2.0000e-06
37/40	67ms	0.3083	0.8852	0.2704	0.9000	2.0000e-06
38/40	63ms	0.3114	0.8767	0.2859	0.8911	2.0000e-06
39/40	63ms	0.3112	0.8881	0.2695	0.9000	2.0000e-06
40/40	62ms	0.3002	0.8871	0.2727	0.8911	2.0000e-06

Hal ini dapat mengindikasikan bahwa model telah mencapai batas kemampuannya dan tidak mampu mempelajari pola yang lebih kompleks dalam

dataset. Dalam kesimpulannya, meskipun model ini berhasil mencapai tingkat akurasi yang relatif baik dalam melakukan klasifikasi dataset CXR, ada potensi overfitting.

4.3.1.3. Training Validation Validation 20 Epoch

Diperoleh hasil akurasi dan loss pada data pelatihan dan data validasi. Detailnya dapat dilihat pada Gambar 4.8 untuk skenario (a) tiga kelas dan (b) dua kelas.



(a) (b)

Gambar 4.8 ResNet50 20 Epoch. (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.8 dan tabel 4.12 dapat dilihat hasil pada klasifikasi 3 kelas, sehingga dapat disimpulkan bahwa model mengalami peningkatan performa seiring dengan peningkatan epoch. Pada epoch awal, model mencapai akurasi training sebesar 57,97% dan validation accuracy sebesar 73,78%. Namun, setelah 20 epoch, akurasi training meningkat menjadi 80,13% dan validation accuracy mencapai 80,44%. Selama training, terlihat adanya penurunan nilai loss baik pada data training maupun data validation, menunjukkan bahwa model berhasil

menyesuaikan diri dengan data dan mampu melakukan klasifikasi lebih baik. Hal ini juga dapat dilihat dari peningkatan nilai validation accuracy yang menunjukkan kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya.

Tabel 4.12 Hasil ResNet50 20 Epoch 3 Kelas

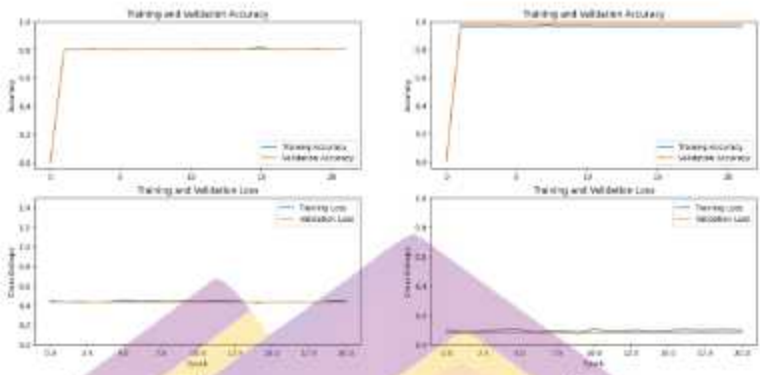
Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/20	74ms	1.1225	0.5797	0.5873	0.7378	0.0010
2/20	34ms	0.7216	0.7149	0.5270	0.7689	0.0010
3/20	34ms	0.6066	0.7476	0.4991	0.7763	0.0010
4/20	35ms	0.5854	0.7511	0.4788	0.7822	0.0010
5/20	34ms	0.5361	0.7667	0.4675	0.8089	0.0010
6/20	32ms	0.5259	0.7562	0.4618	0.8015	0.0010
7/20	33ms	0.5068	0.7794	0.4459	0.8059	0.0010
8/20	34ms	0.5083	0.7787	0.4438	0.8119	0.0010
9/20	32ms	0.5112	0.7733	0.4585	0.7926	0.0010
10/20	32ms	0.4775	0.7879	0.4423	0.7985	0.0010
11/20	32ms	0.4713	0.7860	0.4498	0.8030	0.0010
12/20	34ms	0.4819	0.7848	0.4340	0.8148	0.0010
13/20	35ms	0.4505	0.8048	0.4357	0.8237	0.0010
14/20	34ms	0.4477	0.7971	0.4295	0.8252	0.0010
15/20	32ms	0.4491	0.7978	0.4308	0.8133	0.0010
16/20	32ms	0.4494	0.8006	0.4367	0.8044	0.0010
17/20	32ms	0.4439	0.7984	0.4351	0.8015	0.0010
18/20	32ms	0.4306	0.8060	0.4240	0.8059	0.0010
19/20	32ms	0.4355	0.8060	0.4377	0.7896	0.0010
20/20	32ms	0.4425	0.8013	0.4334	0.8044	0.0010

Pada 2 kelas, dapat disimpulkan bahwa model mencapai peningkatan yang signifikan dalam akurasi model dapat dilihat dari epoch 1 hingga epoch 8, dengan akurasi training naik dari 77,14% menjadi 94,48% dan akurasi validation naik dari 92,44% menjadi 96,44%. Setelah itu, peningkatan akurasi yang lebih lambat terjadi hingga epoch 20, dengan akurasi training mencapai 96,10% dan akurasi validation mencapai 97,78% yang dapat dilihat pada tabel 4.13.

Tabel 4.13 Hasil ResNet50 20 Epoch 2 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/20	97ms	0.5985	0.7714	0.1845	0.9244	0.0010
2/20	39ms	0.2518	0.9100	0.7379	0.9489	0.0010
3/20	35ms	0.2042	0.9252	0.1368	0.9378	0.0010
4/20	35ms	0.1789	0.9352	0.1239	0.9444	0.0010
5/20	34ms	0.1721	0.9386	0.1126	0.9489	0.0010
6/20	34ms	0.1600	0.9410	0.1162	0.9467	0.0010
7/20	34ms	0.1521	0.9452	0.1087	0.9467	0.0010
8/20	38ms	0.1497	0.9448	0.0920	0.9644	0.0010
9/20	34ms	0.1272	0.9562	0.0922	0.9644	0.0010
10/20	34ms	0.1237	0.9532	0.0965	0.9600	0.0010
11/20	34ms	0.1234	0.9586	0.0948	0.9644	0.0010
12/20	34ms	0.1283	0.9548	0.0927	0.9622	0.0010
13/20	38ms	0.1233	0.9562	0.0867	0.9689	0.0010
14/20	34ms	0.1116	0.9610	0.1027	0.9644	0.0010
15/20	34ms	0.1080	0.9629	0.0889	0.9622	0.0010
16/20	34ms	0.1136	0.9576	0.0922	0.9689	0.0010
17/20	38ms	0.1010	0.9624	0.0797	0.9756	0.0010
18/20	34ms	0.1014	0.9614	0.0765	0.9756	0.0010
19/20	35ms	0.1003	0.9667	0.0814	0.9711	0.0010
20/20	38ms	0.0995	0.9610	0.0760	0.9778	0.0010

Hal ini menunjukkan bahwa model telah mencapai tingkat akurasi yang tinggi dan stabil setelah 20 epoch. Selanjutnya, peneliti melakukan perbaikan menggunakan fine tuning yang berfungsi untuk melakukan penyesuaian learning rate yang diatur dari mulai terkecil sampai terbesar dengan menambahkan 30 epoch tambahan. Hasil dari fine tuning ini dapat dilihat pada Gambar 4.9.



(a) (b)

Gambar 4.9 Fine Tuning ResNet50 20 Epoch. (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Hasil finetuning dapat dilihat pada gambar 4.9 dan tabel 4.14, hasil pada 3 kelas dengan 50 epoch menunjukkan hasil yang cukup memuaskan. Akurasi training mencapai 80,5% dan akurasi validasi 80,4% pada epoch terakhir. Terdapat fluktuasi nilai loss yang tinggi dan meskipun dilakukan penurunan learning rate menggunakan metode ReduceLROnPlateau, tidak ada perubahan signifikan dalam performa model.

Tabel 4.14 Hasil Fine Tuning ResNet50 20 Epoch 3 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
20/50	60ms	0.4447	0.8025	0.4334	0.8044	1.0000e-04
21/50	32ms	0.4361	0.8010	0.4334	0.8044	1.0000e-04
22/50	32ms	0.4403	0.8089	0.4334	0.8044	1.0000e-04
23/50	32ms	0.4348	0.8038	0.4334	0.8044	1.0000e-04
24/50	32ms	0.4394	0.8006	0.4334	0.8044	1.0000e-04
25/50	32ms	0.4514	0.8044	0.4334	0.8044	1.0000e-04
26/50	32ms	0.4422	0.8060	0.4334	0.8044	1.0000e-04
27/50	32ms	0.4471	0.8044	0.4334	0.8044	1.0000e-04
28/50	32ms	0.4409	0.8012	0.4334	0.8044	1.0000e-04
29/50	32ms	0.4451	0.8038	0.4334	0.8044	1.0000e-05
30/50	33ms	0.4429	0.8000	0.4334	0.8044	1.0000e-05

Tabel 4.14 Hasil Fine Tuning ResNet50 20 Epoch 3 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
31/50	32ms	0.4485	0.8016	0.4334	0.8044	1.0000e-05
32/50	32ms	0.4472	0.8000	0.4334	0.8044	1.0000e-05
33/50	32ms	0.4396	0.8076	0.4334	0.8044	1.0000e-05
34/50	33ms	0.4264	0.8152	0.4334	0.8044	1.0000e-05
35/50	33ms	0.4355	0.8013	0.4334	0.8044	1.0000e-05
36/50	33ms	0.4354	0.8070	0.4334	0.8044	1.0000e-05
37/50	33ms	0.4369	0.8035	0.4334	0.8044	2.0000e-06
38/50	33ms	0.4380	0.8086	0.4334	0.8044	2.0000e-06
39/50	32ms	0.4491	0.8035	0.4334	0.8044	2.0000e-06
40/50	33ms	0.4477	0.8048	0.4334	0.8044	2.0000e-06

Sedangkan untuk 2 kelas, dapat disimpulkan bahwa model mencapai akurasi training sebesar 96,33% setelah 20 epoch, sedangkan validation accuracy mencapai 97,78%. Meskipun model telah mencapai tingkat akurasi yang tinggi, tidak ada peningkatan lebih lanjut selama 30 epoch berikutnya. Hal ini ditunjukkan oleh stagnasi nilai loss dan akurasi baik pada data pelatihan maupun data validasi. Pada epoch 28, dilakukan pengurangan learning rate menggunakan ReduceLROnPlateau. Meskipun demikian, tidak ada peningkatan tambahan dalam performa model yang dapat dilihat pada tabel 4.15.

Tabel 4.15 Hasil Fine Tuning ResNet50 20 Epoch 2 Kelas

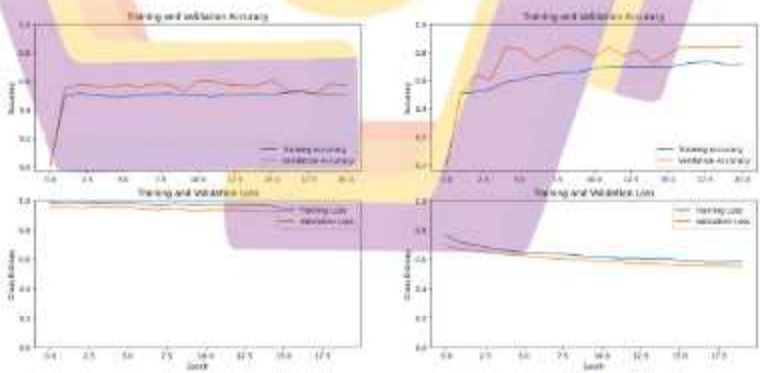
Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
20/50	76ms	0.0987	0.9633	0.0760	0.9778	1.0000e-04
21/50	33ms	0.0893	0.9662	0.0760	0.9778	1.0000e-04
22/50	34ms	0.0910	0.9638	0.0760	0.9778	1.0000e-04
23/50	33ms	0.0963	0.9676	0.0760	0.9778	1.0000e-04
24/50	34ms	0.1009	0.9624	0.0760	0.9778	1.0000e-04
25/50	33ms	0.1009	0.9595	0.0760	0.9778	1.0000e-04
26/50	33ms	0.0867	0.9710	0.0760	0.9778	1.0000e-04
27/50	33ms	0.0937	0.9657	0.0760	0.9778	1.0000e-04
28/50	35ms	0.0941	0.9662	0.0760	0.9778	1.0000e-04

Tabel 4.15 Hasil Fine Tuning ResNet50 20 Epoch 2 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
29/50	34ms	0.0860	0.9671	0.0760	0.9778	1.0000e-05
30/50	34ms	0.7037	0.9624	0.0760	0.9778	1.0000e-05
31/50	34ms	0.0913	0.9643	0.0760	0.9778	1.0000e-05
32/50	33ms	0.0932	0.9638	0.0760	0.9778	1.0000e-05
33/50	33ms	0.0986	0.9643	0.0760	0.9778	1.0000e-05
34/50	33ms	0.0905	0.9629	0.0760	0.9778	1.0000e-05
35/50	34ms	0.0956	0.9595	0.0760	0.9778	1.0000e-05
36/50	34ms	0.1012	0.9638	0.0760	0.9778	1.0000e-05
37/50	34ms	0.0978	0.9629	0.0760	0.9778	2.0000e-06
38/50	35ms	0.0986	0.9590	0.0760	0.9778	2.0000e-06
39/50	35ms	0.1010	0.9624	0.0760	0.9778	2.0000e-06
40/50	39ms	0.0967	0.9633	0.0760	0.9778	2.0000e-06

4.3.1.4. Training Validation 20 Epoch (CLAHE – White Balance)

Diperoleh hasil akurasi dan loss pada data pelatihan dan data validasi. Detailnya dapat dilihat pada Gambar 4.10 untuk skenario (a) tiga kelas dan (b) dua kelas.



(a) (b)

Gambar 4.10 ResNet50 20 Epoch (CLAHE – White Balance). (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.10 dan tabel 16 dapat dilihat hasil klasifikasi dengan 3 kelas, model menghasilkan loss sekitar 95,67% dan akurasi sekitar 50,67% pada data training. Sedangkan pada data validation, mendapatkan hasil loss sekitar 92,99% dan akurasi sekitar 57,04%. Hasil tersebut menunjukkan bahwa model belum mencapai performa optimal. Perbedaan yang signifikan antara akurasi pada data pelatihan dan data validasi, menunjukkan adanya overfitting. Akurasi yang relatif rendah menunjukkan bahwa model masih belum mampu menggeneralisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya.

Tabel 4.16 Hasil ResNet50 20 Epoch 3 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/20	67ms	0.9873	0.4949	0.9533	0.5481	0.0010
2/20	53ms	0.9790	0.5146	0.9528	0.5748	0.0010
3/20	50ms	0.9811	0.5025	0.9447	0.5659	0.0010
4/20	50ms	0.9845	0.4927	0.9608	0.5570	0.0010
5/20	53ms	0.9809	0.4927	0.9522	0.5778	0.0010
6/20	50ms	0.9814	0.4981	0.9547	0.5585	0.0010
7/20	51ms	0.9753	0.5089	0.9383	0.5778	0.0010
8/20	50ms	0.9631	0.5121	0.9340	0.5778	0.0010
9/20	50ms	0.9778	0.5038	0.9460	0.5244	0.0010
10/20	53ms	0.9693	0.5025	0.9300	0.5985	0.0010
11/20	51ms	0.9410	0.4917	0.9339	0.5985	0.0010
12/20	50ms	0.9423	0.5070	0.9348	0.5748	0.0010
13/20	50ms	0.9693	0.5073	0.9345	0.5689	0.0010
14/20	50ms	0.9693	0.5121	0.9286	0.5659	0.0010
15/20	53ms	0.9660	0.5048	0.9224	0.6089	0.0010
16/20	50ms	0.9506	0.5197	0.9303	0.5348	0.0010
17/20	50ms	0.9525	0.5251	0.9392	0.5319	0.0010
18/20	49ms	0.9612	0.5083	0.9313	0.5126	0.0010
19/20	49ms	0.9643	0.5083	0.9224	0.5793	0.0010
20/20	49ms	0.9567	0.5067	0.9299	0.5724	0.0010

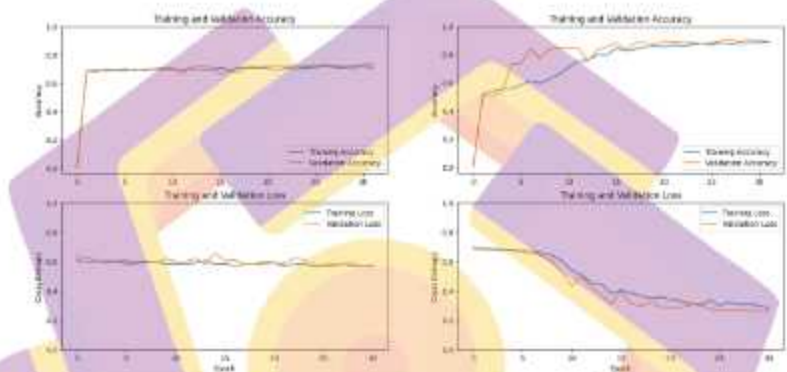
Pada klasifikasi dengan 2 kelas, model mencapai loss sekitar 58,30% dan akurasi sekitar 71,86% pada data training. Sedangkan pada data validasi, model mendapatkan hasil loss sekitar 55,28% dan akurasi sekitar 83,33%. Terjadi peningkatan performa model dari epoch. Meskipun terdapat perbedaan yang signifikan antara akurasi pada data training dan data validasi, model masih mampu menggeneralisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya, dengan akurasi yang tinggi dan dapat dilihat pada tabel 4.17.

Tabel 4.17 Hasil ResNet50 20 Epoch 2 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/20	62ms	0.7635	0.5086	0.6903	0.5000	0.0010
2/20	57ms	0.7096	0.5176	0.6695	0.6422	0.0010
3/20	55ms	0.6936	0.5376	0.6562	0.6022	0.0010
4/20	55ms	0.6674	0.5886	0.6411	0.8156	0.0010
5/20	58ms	0.6573	0.6043	0.6297	0.8222	0.0010
6/20	52ms	0.6451	0.6367	0.6240	0.7378	0.0010
7/20	45ms	0.6412	0.6414	0.6126	0.7933	0.0010
8/20	43ms	0.6351	0.6562	0.6026	0.8400	0.0010
9/20	45ms	0.6317	0.6610	0.5959	0.8222	0.0010
10/20	43ms	0.6182	0.6886	0.5956	0.7689	0.0010
11/20	43ms	0.6144	0.6938	0.5804	0.8356	0.0010
12/20	45ms	0.6045	0.7005	0.5848	0.7778	0.0010
13/20	43ms	0.6060	0.6938	0.5694	0.8089	0.0010
14/20	45ms	0.6014	0.6971	0.5723	0.7244	0.0010
15/20	43ms	0.6058	0.6924	0.5682	0.7889	0.0010
16/20	43ms	0.5895	0.7162	0.5559	0.8400	0.0010
17/20	43ms	0.5852	0.7295	0.5545	0.8333	1.0000e-04
18/20	46ms	0.5779	0.7310	0.5540	0.8333	1.0000e-04
19/20	43ms	0.5796	0.7129	0.5531	0.8378	1.0000e-04
20/20	45ms	0.5830	0.7186	0.5528	0.8333	1.0000e-04

Meskipun demikian, terdapat potensi untuk meningkatkan performa model lebih lanjut. Upaya untuk peningkatan performa dapat dilakukan dengan melakukan

penyesuaian pada arsitektur model mengurangi overfitting. Selanjutnya, peneliti melakukan perbaikan menggunakan fine tuning yang berfungsi untuk melakukan penyesuaian learning rate yang diatur dari mulai terkecil sampai terbesar dengan menambahkan 30 epoch tambahan. Hasil dari finetuning ini dapat dilihat pada Gambar 4.11.



(a) (b)

Gambar 4.11 Fine Tuning ResNet50 20 Epoch (CLAHE – White Balance). (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.11 dan tabel 4.18 dapat dilihat bahwa fine tuning klasifikasi 3 kelas mencapai akurasi sekitar 70% pada data training dan sekitar 73% pada data validasi. Hasil loss pada data training berkisar sekitar 57%, sedangkan loss pada data validasi berkisar sekitar 56%. Pada awalnya, model menunjukkan peningkatan yang cepat dalam akurasi dan penurunan loss pada setiap epoch. Namun, setelah beberapa epoch, peningkatan akurasi dan penurunan loss menjadi lebih lambat dan cenderung stabil. Terdapat beberapa fluktuasi dalam akurasi dan loss pada setiap epoch, tetapi secara keseluruhan, model tersebut menunjukkan kemampuan yang baik dalam mengklasifikasikan.

Tabel 4.18 Hasil Fine Tuning ResNet50 20 Epoch 3 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
20/50	78ms	0.6087	0.6844	0.6320	0.6889	1.0000e-05
21/50	62ms	0.5960	0.6898	0.6295	0.6770	1.0000e-05
22/50	65ms	0.6012	0.6867	0.6035	0.6978	1.0000e-05
23/50	62ms	0.5944	0.6879	0.6043	0.6919	1.0000e-05
24/50	66ms	0.5978	0.6867	0.6042	0.7022	1.0000e-05
25/50	62ms	0.5862	0.6940	0.6068	0.6859	1.0000e-05
26/50	62ms	0.5879	0.6908	0.5996	0.6993	1.0000e-05
27/50	62ms	0.5925	0.6965	0.6039	0.7007	1.0000e-05
28/50	65ms	0.5929	0.6911	0.5943	0.7081	1.0000e-05
29/50	63ms	0.5838	0.6975	0.6182	0.7081	1.0000e-05
30/50	62ms	0.5850	0.6971	0.6017	0.6696	1.0000e-05
31/50	64ms	0.5823	0.7013	0.5993	0.7111	1.0000e-05
32/50	65ms	0.5897	0.6962	0.6210	0.7185	1.0000e-05
33/50	62ms	0.6070	0.6924	0.5812	0.7141	1.0000e-05
34/50	62ms	0.5805	0.7086	0.6582	0.6637	1.0000e-05
35/50	62ms	0.5854	0.6940	0.6028	0.6844	1.0000e-05
36/50	62ms	0.5712	0.7067	0.6115	0.6756	1.0000e-05
37/50	65ms	0.5805	0.7067	0.5804	0.7215	1.0000e-05
38/50	62ms	0.5815	0.7063	0.5810	0.7126	1.0000e-05
39/50	62ms	0.5940	0.7044	0.5815	0.7215	1.0000e-05
40/50	62ms	0.5923	0.6943	0.5908	0.7215	1.0000e-05
41/50	62ms	0.5687	0.7067	0.5829	0.7170	1.0000e-05
42/50	65ms	0.5800	0.6997	0.6205	0.7244	1.0000e-05
43/50	62ms	0.5839	0.7083	0.6129	0.7200	1.0000e-05
44/50	62ms	0.5662	0.7114	0.5746	0.7230	1.0000e-05
45/50	64ms	0.5697	0.7171	0.5762	0.7304	1.0000e-05
46/50	62ms	0.5682	0.7117	0.5859	0.7230	1.0000e-05
47/50	62ms	0.5837	0.7073	0.5783	0.7185	1.0000e-05
48/50	62ms	0.5666	0.7241	0.5891	0.6978	1.0000e-05
49/50	64ms	0.5657	0.7140	0.5657	0.7319	1.0000e-05
50/50	62ms	0.5739	0.7095	0.5635	0.7319	1.0000e-05

Selanjutnya pada gambar tersebut dapat dilihat hasil fine tuning klasifikasi 2 kelas menunjukkan peningkatan dalam akurasi selama epoch awal, tetapi

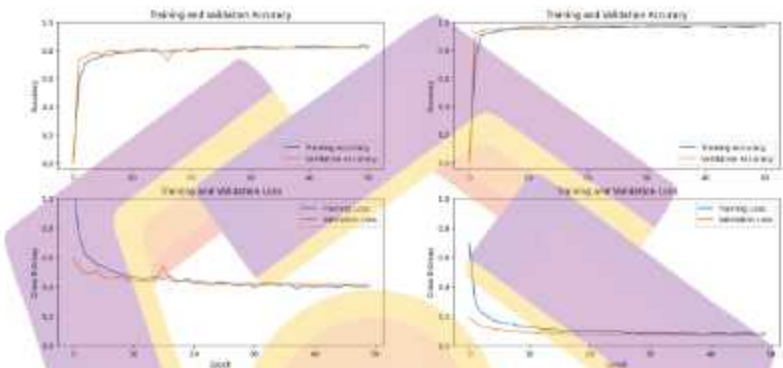
kemudian meningkat dengan lebih lambat. Pada akhir pelatihan, model mencapai akurasi sekitar 89% pada data training dan sekitar 90% pada data validasi yang dapat dilihat pada tabel 4.19.

Tabel 4.19 Hasil Fine Tuning ResNet50 20 Epoch 2 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
20/50	61ms	0.6900	0.5171	0.6857	0.5000	1.0000e-05
21/50	57ms	0.6868	0.5390	0.6825	0.5067	1.0000e-05
22/50	43ms	0.6849	0.5529	0.6794	0.5289	1.0000e-05
23/50	43ms	0.6827	0.5557	0.6759	0.7289	1.0000e-05
24/50	54ms	0.6799	0.5843	0.6717	0.7378	1.0000e-05
25/50	47ms	0.6728	0.6129	0.6643	0.8400	1.0000e-05
26/50	43ms	0.6678	0.5952	0.6521	0.7733	1.0000e-05
27/50	50ms	0.6566	0.6243	0.6359	0.8378	1.0000e-05
28/50	44ms	0.6430	0.6614	0.6026	0.8444	1.0000e-05
29/50	40ms	0.6038	0.7086	0.5269	0.8422	1.0000e-05
30/50	46ms	0.5398	0.7519	0.4320	0.8467	1.0000e-05
31/50	43ms	0.5084	0.7590	0. 5035	0.7533	1.0000e-05
32/50	43ms	0.4522	0.7981	0.4213	0.8333	1.0000e-05
33/50	43ms	0.4452	0.7943	0.3726	0.8533	1.0000e-05
34/50	48ms	0.3802	0.8452	0.3100	0.8844	1.0000e-05
35/50	46ms	0.4049	0.8271	0.3974	0.8333	1.0000e-05
36/50	43ms	0.3856	0.8386	0.3117	0.8867	1.0000e-05
37/50	43ms	0.3654	0.8519	0.2976	0.8867	1.0000e-05
38/50	43ms	0.3469	0.8619	0.3207	0.8867	1.0000e-05
39/50	50ms	0.3590	0.8562	0.2890	0.8933	1.0000e-05
40/50	46ms	0.3407	0.8690	0.2848	0.8911	1.0000e-05
41/50	43ms	0.3313	0.8614	0.2854	0.8867	1.0000e-05
42/50	43ms	0.3132	0.8705	0.3049	0.8867	1.0000e-05
43/50	46ms	0.3020	0.8871	0.3098	0.8733	1.0000e-05
44/50	42ms	0.3360	0.8671	0.2780	0.8867	1.0000e-05
45/50	43ms	0.2989	0.8829	0.2669	0.8956	1.0000e-05
46/50	43ms	0.3219	0.8748	0.2675	0.9067	1.0000e-05
47/50	45ms	0.3043	0.8805	0.2756	0.8889	1.0000e-05
48/50	43ms	0.3083	0.8862	0.2634	0.9089	1.0000e-05
49/50	43ms	0.2997	0.8848	0.2636	0.8933	1.0000e-05
50/50	43ms	0.2819	0.8910	0.2582	0.8956	1.0000e-05

4.3.1.5. Training Validation 50 Epoch

Diperoleh hasil akurasi dan loss pada data pelatihan dan data validasi. Informasi terperinci dapat dilihat pada Gambar 4.12 untuk skenario (a) dengan tiga kelas dan (b) dengan dua kelas.



(a) (b)

Gambar 4.12 ResNet50 50 Epoch. (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.12 dan tabel 4.20 dapat dilihat bahwa training klasifikasi 3 kelas, awalnya mencapai akurasi sekitar 58,79% dan loss sekitar 109,88% pada training. Selama pelatihan, peneliti melihat peningkatan dalam akurasi dan penurunan dalam loss pada training dan validation. Hingga pada akhirnya, model mencapai akurasi sekitar 82,22% dan loss sekitar 41,69% pada training, serta akurasi sekitar 81,78% dan loss sekitar 41,69% pada set validasi. Hal ini menunjukkan bahwa model secara bertahap meningkat dalam kemampuannya untuk mengklasifikasikan data dengan benar. Penggunaan pengurangan learning rate yang adaptif dengan menggunakan metode ReduceLROnPlateau juga membantu dalam meningkatkan kinerja model.

Tabel 4.20 Hasil ResNet50 50 Epoch 3 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/50	76ms	1.0988	0.5879	0.5962	0.7363	0.0010
2/50	36ms	0.7386	0.7032	0.5306	0.7526	0.0010
3/50	36ms	0.6174	0.7349	0.4897	0.7807	0.0010
4/50	35ms	0.5871	0.7451	0.4917	0.7822	0.0010
5/50	33ms	0.5460	0.7619	0.5066	0.7689	0.0010
6/50	36ms	0.5408	0.7622	0.4640	0.8000	0.0010
7/50	33ms	0.5085	0.7781	0.4579	0.8000	0.0010
8/50	33ms	0.4993	0.7810	0.4620	0.7926	0.0010
9/50	33ms	0.4849	0.7903	0.4733	0.7807	0.0010
10/50	36ms	0.4715	0.7889	0.4424	0.8030	0.0010
11/50	33ms	0.4694	0.7952	0.4452	0.8030	0.0010
12/50	35ms	0.4543	0.8032	0.4402	0.8089	0.0010
13/50	33ms	0.4521	0.8003	0.4425	0.8000	0.0010
14/50	36ms	0.4631	0.7930	0.4410	0.8104	0.0010
15/50	33ms	0.4627	0.7883	0.4687	0.7793	0.0010
16/50	33ms	0.4502	0.7959	0.5414	0.7274	0.0010
17/50	33ms	0.4737	0.7914	0.4400	0.8059	0.0010
18/50	33ms	0.4401	0.8016	0.4302	0.7985	0.0010
19/50	33ms	0.4421	0.8057	0.4417	0.8089	0.0010
20/50	33ms	0.4496	0.8000	0.4390	0.7867	0.0010
21/50	33ms	0.4299	0.8130	0.4235	0.8044	0.0010
22/50	33ms	0.4389	0.8079	0.4201	0.8089	0.0010
23/50	36ms	0.4199	0.8108	0.4229	0.8148	1.0000e-04
24/50	33ms	0.4154	0.8121	0.4199	0.8104	1.0000e-04
25/50	33ms	0.4279	0.8127	0.4195	0.8133	1.0000e-04
26/50	33ms	0.4254	0.8092	0.4187	0.8044	1.0000e-04
27/50	33ms	0.4204	0.8165	0.4214	0.8148	1.0000e-04
28/50	34ms	0.4085	0.8257	0.4184	0.8074	1.0000e-04
29/50	34ms	0.4109	0.8235	0.4186	0.8119	1.0000e-04
30/50	35ms	0.4214	0.8206	0.4224	0.8193	1.0000e-04
31/50	33ms	0.3962	0.8292	0.4190	0.8148	1.0000e-04
32/50	33ms	0.4157	0.8219	0.4193	0.8163	1.0000e-04
33/50	33ms	0.4111	0.8216	0.4263	0.8163	1.0000e-04
34/50	33ms	0.4250	0.8165	0.4186	0.8104	1.0000e-04
35/50	33ms	0.4057	0.8194	0.4194	0.8163	1.0000e-04
36/50	33ms	0.4189	0.8178	0.4199	0.8178	1.0000e-04

Tabel 4.20 Hasil ResNet50 50 Epoch 3 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
37/50	33ms	0.4138	0.8768	0.4194	0.8178	1.0000e-04
38/50	33ms	0.3873	0.8305	0.4188	0.8178	1.0000e-04
39/50	33ms	0.4056	0.8235	0.4184	0.8163	1.0000e-05
40/50	33ms	0.4013	0.8276	0.4180	0.8148	1.0000e-05
41/50	33ms	0.3989	0.8305	0.4180	0.8148	1.0000e-05
42/50	35ms	0.4027	0.8267	0.4175	0.8207	1.0000e-05
43/50	33ms	0.4020	0.8263	0.4175	0.8193	1.0000e-05
44/50	33ms	0.4030	0.8238	0.4172	0.8193	1.0000e-05
45/50	33ms	0.3943	0.8222	0.4176	0.8193	1.0000e-05
46/50	33ms	0.4102	0.8213	0.4169	0.8178	1.0000e-05
47/50	33ms	0.4023	0.8222	0.4169	0.8178	1.0000e-05
48/50	33ms	0.4018	0.8225	0.4171	0.8193	1.0000e-05
49/50	33ms	0.4004	0.8352	0.4174	0.8178	1.0000e-05
50/50	34ms	0.4029	0.8267	0.4172	0.8193	1.0000e-05

Sedangkan pada 2 kelas, awalnya tingkat akurasi model hanya sekitar 73,38% dengan nilai loss sebesar 69,65% pada epoch pertama. Namun, seiring berjalannya waktu, model mengalami peningkatan yang signifikan dalam kinerja. Pada akhir training, tingkat akurasi model mencapai 97,48% dengan nilai loss sebesar 7,58% pada epoch terakhir sesuai dengan tabel dapat dilihat pada tabel 4.21.

Tabel 4.21 Hasil ResNet50 50 Epoch 2 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/50	95ms	0.6965	0.7338	0.1890	0.9222	0.0010
2/50	37ms	0.2823	0.9062	0.1538	0.9378	0.0010
3/50	37ms	0.2225	0.9195	0.1300	0.9467	0.0010
4/50	33ms	0.2015	0.9262	0.1315	0.9422	0.0010
5/50	36ms	0.1683	0.9400	0.1085	0.9489	0.0010
6/50	37ms	0.1570	0.9443	0.1109	0.9511	0.0010
7/50	37ms	0.1459	0.9462	0.0998	0.9533	0.0010
8/50	36ms	0.1466	0.9476	0.0970	0.9622	0.0010
9/50	37ms	0.1322	0.9529	0.0943	0.9644	0.0010
10/50	33ms	0.1327	0.9529	0.0896	0.9644	0.0010

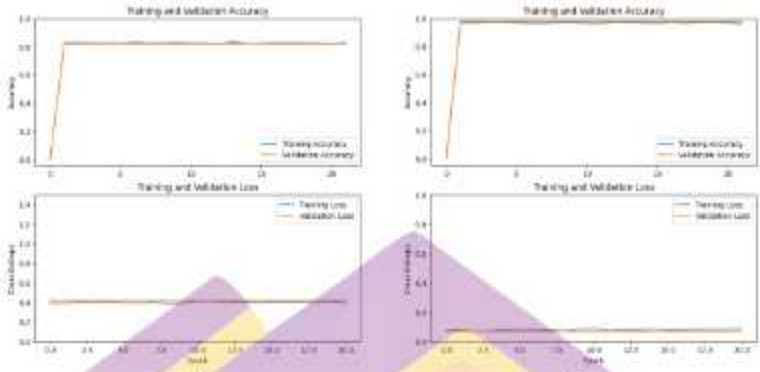
Tabel 4.21 Hasil ResNet50 50 Epoch 2 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/50	95ms	0.6965	0.7338	0.1890	0.9222	0.0010
2/50	37ms	0.2823	0.9062	0.1538	0.9378	0.0010
3/50	37ms	0.2225	0.9195	0.1300	0.9467	0.0010
4/50	33ms	0.2015	0.9262	0.1315	0.9422	0.0010
5/50	36ms	0.1683	0.9400	0.1085	0.9489	0.0010
6/50	37ms	0.1570	0.9443	0.1109	0.9511	0.0010
7/50	37ms	0.1459	0.9462	0.0998	0.9533	0.0010
8/50	36ms	0.1466	0.9476	0.0970	0.9622	0.0010
9/50	37ms	0.1322	0.9529	0.0943	0.9644	0.0010
10/50	33ms	0.1327	0.9529	0.0896	0.9644	0.0010
11/50	37ms	0.1273	0.9567	0.0904	0.9689	0.0010
12/50	37ms	0.1209	0.9581	0.0858	0.9711	0.0010
13/50	33ms	0.1132	0.9567	0.0837	0.9622	0.0010
14/50	33ms	0.1117	0.9552	0.0932	0.9578	0.0010
15/50	37ms	0.1166	0.9600	0.0793	0.9756	0.0010
16/50	33ms	0.0979	0.9686	0.0865	0.9622	0.0010
17/50	33ms	0.1006	0.9610	0.0817	0.9600	0.0010
18/50	33ms	0.1058	0.9619	0.0778	0.9756	0.0010
19/50	33ms	0.0923	0.9690	0.0779	0.9756	0.0010
20/50	37ms	0.0967	0.9624	0.0759	0.9778	0.0010
21/50	33ms	0.0991	0.9629	0.0764	0.9756	0.0010
22/50	33ms	0.1014	0.9619	0.0769	0.9778	0.0010
23/50	33ms	0.0906	0.9695	0.0761	0.9733	0.0010
24/50	33ms	0.0929	0.9652	0.0741	0.9778	0.0010
25/50	33ms	0.0963	0.9648	0.0783	0.9733	0.0010
26/50	33ms	0.0940	0.9662	0.0751	0.9756	0.0010
27/50	33ms	0.0848	0.9686	0.0827	0.9644	0.0010
28/50	33ms	0.0917	0.9676	0.0744	0.9756	0.0010
29/50	33ms	0.0821	0.9714	0.0723	0.9733	1.0000e-04
30/50	33ms	0.0866	0.9690	0.0717	0.9756	1.0000e-04
31/50	37ms	0.0828	0.9700	0.0710	0.9822	1.0000e-04
32/50	33ms	0.0893	0.9714	0.0710	0.9778	1.0000e-04
33/50	33ms	0.0856	0.9695	0.0709	0.9778	1.0000e-04
34/50	33ms	0.0860	0.9705	0.0710	0.9800	1.0000e-04
35/50	33ms	0.0807	0.9700	0.0707	0.9800	1.0000e-04
36/50	33ms	0.0848	0.9667	0.0711	0.9756	1.0000e-04

Tabel 4.21 Hasil ResNet50 50 Epoch 2 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
37/50	33ms	0.0911	0.9652	0.0714	0.9756	1.0000e-04
38/50	33ms	0.0802	0.9743	0.0718	0.9756	1.0000e-04
39/50	33ms	0.0804	0.9733	0.0731	0.9756	1.0000e-04
40/50	33ms	0.0758	0.9748	0.0722	0.9733	1.0000e-05
41/50	33ms	0.0811	0.9700	0.0718	0.9756	1.0000e-05
42/50	33ms	0.0839	0.9695	0.0717	0.9756	1.0000e-05
43/50	33ms	0.0884	0.9643	0.0714	0.9756	1.0000e-05
44/50	33ms	0.0762	0.9724	0.0714	0.9756	1.0000e-05
45/50	33ms	0.0827	0.9690	0.0712	0.9778	1.0000e-05
46/50	33ms	0.0799	0.9662	0.0711	0.9778	1.0000e-05
47/50	33ms	0.0893	0.9671	0.0712	0. 9778	1.0000e-05
48/50	33ms	0.0818	0.9667	0.0712	0. 9778	2.0000e-06
49/50	33ms	0.0734	0.9733	0.0712	0. 9778	2.0000e-06
50/50	33ms	0.0830	0.9700	0.0712	0. 9778	2.0000e-06

Selama pelatihan, learning rate juga secara otomatis disesuaikan dengan menggunakan teknik ReduceLROnPlateau untuk membantu model mencapai konvergensi yang lebih baik. Teknik ini mengurangi learning rate saat kemajuan dalam validasi terhenti. Untuk memperbaiki hasil pada model peneliti melakukan perbaikan menggunakan fine tuning yang berfungsi untuk melakukan penyesuaian learning rate yang diatur dari mulai terkecil sampai terbesar dengan menambahkan 30 epoch yang dapat dilihat pada gambar 4.13.



(a) (b)

Gambar 4.13 Fine Tuning ResNet50 50 Epoch. (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Hasil finetuning pada 3 kelas dapat dilihat pada gambar 4.13 dan tabel 4.22, pada awal training model mencapai tingkat akurasi sekitar 82,63% pada data training dan 81,93% pada data validation. Tahap ini, model mengalami penurunan learning rate menggunakan teknik ReduceLROnPlateau. Setelah penurunan learning rate, performa model tidak banyak berubah. Model tetap menghasilkan akurasi sekitar 81,93% pada data validasi hingga akhir training. Hasil loss model menghasilkan sekitar 41,72% pada data validasi.

Tabel 4.22 Hasil Fine Tuning ResNet50 50 Epoch 3 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
50/80	61ms	0.3917	0.8263	0.4172	0.8193	1.0000e-04
51/80	33ms	0.3940	0.8298	0.4172	0.8193	1.0000e-04
52/80	33ms	0.4054	0.8260	0.4172	0.8193	1.0000e-04
53/80	33ms	0.4010	0.8263	0.4172	0.8193	1.0000e-04
54/80	33ms	0.4082	0.8225	0.4172	0.8193	1.0000e-04
55/80	33ms	0.3976	0.8321	0.4172	0.8193	1.0000e-04
56/80	33ms	0.3950	0.8251	0.4172	0.8193	1.0000e-04
57/80	34ms	0.4039	0.8270	0.4172	0.8193	1.0000e-04
58/80	33ms	0.3869	0.8295	0.4172	0.8193	1.0000e-04

Tabel 4.22 Hasil Fine Tuning ResNet50 50 Epoch 3 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
59/80	33ms	0.3922	0.8257	0.4172	0.8193	1.0000e-05
60/80	33ms	0.4125	0.8254	0.4172	0.8193	1.0000e-05
61/80	33ms	0.4114	0.8159	0.4172	0.8193	1.0000e-05
62/80	33ms	0.4017	0.8375	0.4172	0.8193	1.0000e-05
63/80	33ms	0.4071	0.8203	0.4172	0.8193	1.0000e-05
64/80	33ms	0.4002	0.8222	0.4172	0.8193	1.0000e-05
65/80	33ms	0.4048	0.8286	0.4172	0.8193	1.0000e-05
66/80	33ms	0.4048	0.8286	0.4172	0.8193	1.0000e-05
67/80	33ms	0.4046	0.8270	0.4172	0.8193	2.0000e-06
68/80	33ms	0.3976	0.8267	0.4172	0.8193	2.0000e-06
69/80	33ms	0.4087	0.8175	0.4172	0.8193	2.0000e-06
70/80	36ms	0.3934	0.8314	0.4172	0.8193	2.0000e-06

Sedangkan untuk 2 kelas, Hasil training tersebut menampilkan bahwa selama proses training, nilai loss dan akurasi dievaluasi untuk setiap epoch. Selama berjalannya epoch, terlihat adanya penurunan nilai loss dan peningkatan nilai akurasi secara bertahap. Pada epoch terakhir, diperoleh loss sebesar 7,12% dan akurasi sebesar 97,78% pada data validasi. Selain itu, pada beberapa epoch, learning rate juga mengalami penyesuaian menggunakan teknik ReduceLROnPlateau. Teknik ini memungkinkan pengurangan learning rate secara otomatis ketika peningkatan performa model tidak signifikan yang dapat dilihat pada tabel 4.23.

Tabel 4.23 Hasil Fine Tuning ResNet50 50 Epoch 2 Kelas

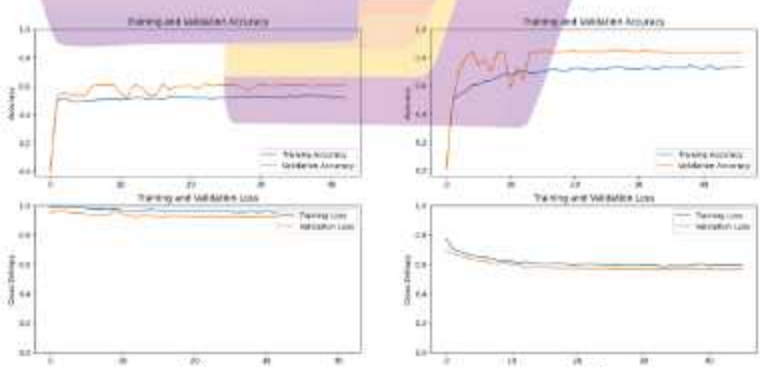
Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
50/80	76ms	0.0753	0.9705	0.0712	0.9778	1.0000e-04
51/80	33ms	0.0779	0.9705	0.0712	0.9778	1.0000e-04
52/80	33ms	0.0704	0.9748	0.0712	0.9778	1.0000e-04
53/80	33ms	0.0719	0.9743	0.0712	0.9778	1.0000e-04
54/80	34ms	0.0818	0.9719	0.0712	0.9778	1.0000e-04

Tabel 4.23 Hasil Fine Tuning ResNet50 50 Epoch 2 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
55/80	33ms	0.0829	0.9676	0.0712	0.9778	1.0000e-04
56/80	33ms	0.0814	0.9676	0.0712	0.9778	1.0000e-04
57/80	32ms	0.0791	0.9714	0.0712	0.9778	1.0000e-04
58/80	33ms	0.0716	0.9719	0.0712	0.9778	1.0000e-04
59/80	33ms	0.0859	0.9676	0.0712	0.9778	1.0000e-05
60/80	33ms	0.0881	0.9667	0.0712	0.9778	1.0000e-05
61/80	33ms	0.0735	0.9748	0.0712	0.9778	1.0000e-05
62/80	33ms	0.0810	0.9724	0.0712	0.9778	1.0000e-05
63/80	33ms	0.0786	0.9705	0.0712	0.9778	1.0000e-05
64/80	33ms	0.0876	0.9676	0.0712	0.9778	1.0000e-05
65/80	33ms	0.0771	0.9748	0.0712	0.9778	1.0000e-05
66/80	33ms	0.0802	0.9690	0.0712	0.9778	1.0000e-05
67/80	33ms	0.0826	0.9733	0.0712	0.9778	2.0000e-06
68/80	33ms	0.0832	0.9733	0.0712	0.9778	2.0000e-06
69/80	33ms	0.0836	0.9714	0.0712	0.9778	2.0000e-06
70/80	38ms	0.0895	0.9652	0.0712	0.9778	2.0000e-06

4.3.1.6. Training Validation 50 Epoch (CLAHE – White Balance)

Diperoleh hasil akurasi dan loss pada data pelatihan dan data validasi. Informasi terperinci dapat dilihat pada Gambar 4.14 untuk skenario (a) dengan tiga kelas dan (b) dengan dua kelas.



(a) (b)

Gambar 4.14 ResNet50 50 Epoch (CLAHE – White Balance). (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.14 dan tabel 4.24 dapat dilihat hasil klasifikasi menggunakan 3 kelas, pada awalnya akurasi training sekitar 49,84% dengan loss sebesar 99,07%. Akurasi validasi awalnya sekitar 52,44% dengan loss 95,12%. Selama proses training, terjadi fluktuasi dalam akurasi dan loss pada kedua dataset. Setelah 50 epoch, model mencapai akurasi training sekitar 27% dengan loss 95,39%, sedangkan akurasi validasi sekitar 60,44% dengan loss 92,25%. Terdapat penurunan learning rate selama pelatihan untuk meningkatkan performa model.

Tabel 4.24 Hasil ResNet50 50 Epoch 3 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/50	54ms	0.9907	0.4984	0.9512	0.5244	0.0010
2/50	51ms	0.9874	0.5143	0.9569	0.5570	0.0010
3/50	48ms	0.9812	0.4917	0.9628	0.5289	0.0010
4/50	48ms	0.9891	0.4886	0.9462	0.5422	0.0010
5/50	48ms	0.9878	0.4968	0.9499	0.5289	0.0010
6/50	51ms	0.9789	0.4990	0.9404	0.6000	0.0010
7/50	51ms	0.9730	0.5048	0.9342	0.6119	0.0010
8/50	48ms	0.9741	0.5060	0.9334	0.6059	0.0010
9/50	49ms	0.9711	0.5076	0.9321	0.6104	0.0010
10/50	48ms	0.9747	0.5041	0.9609	0.5437	0.0010
11/50	48ms	0.9672	0.5073	0.9419	0.5185	0.0010
12/50	48ms	0.9635	0.5171	0.9293	0.6104	0.0010
13/50	48ms	0.9577	0.5175	0.9241	0.5911	0.0010
14/50	49ms	0.9673	0.5070	0.9291	0.5348	0.0010
15/50	48ms	0.9715	0.5140	0.9299	0.5348	0.0010
16/50	50ms	0.9654	0.5051	0.9228	0.6148	1.0000e-04
17/50	48ms	0.9542	0.5260	0.9242	0.5733	1.0000e-04
18/50	48ms	0.9593	0.5270	0.9268	0.5985	1.0000e-04
19/50	48ms	0.9649	0.5203	0.9257	0.5985	1.0000e-04
20/50	48ms	0.9560	0.5213	0.9252	0.6030	1.0000e-04
21/50	48ms	0.9624	0.5152	0.9237	0.5793	1.0000e-04
22/50	50ms	0.9583	0.5187	0.9217	0.6189	1.0000e-04

Tabel 4.24 Hasil ResNet50 50 Epoch 3 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
23/50	48ms	0.9641	0.5063	0.9257	0.6015	1.0000e-04
24/50	48ms	0.9593	0.5197	0.9238	0.6074	1.0000e-04
25/50	48ms	0.9606	0.5165	0.9220	0.6089	1.0000e-04
26/50	48ms	0.9553	0.5225	0.9253	0.6074	1.0000e-04
27/50	48ms	0.9526	0.5279	0.9230	0.6000	1.0000e-04
28/50	48ms	0.9541	0.5235	0.9216	0.5644	1.0000e-04
29/50	49ms	0.9563	0.5251	0.9223	0.6000	1.0000e-04
30/50	49ms	0.9520	0.5254	0.9239	0.6119	1.0000e-04
31/50	48ms	0.9592	0.5222	0.9234	0.6030	1.0000e-05
32/50	48ms	0.9524	0.5267	0.9228	0.6015	1.0000e-05
33/50	48ms	0.9552	0.5162	0.9221	0.6074	1.0000e-05
34/50	48ms	0.9533	0.5314	0.9222	0.6059	1.0000e-05
35/50	48ms	0.9526	0.5238	0.9223	0.6074	1.0000e-05
36/50	49ms	0.9535	0.5337	0.9222	0.6089	1.0000e-05
37/50	49ms	0.9530	0.5317	0.9224	0.5985	1.0000e-05
38/50	48ms	0.9542	0.5289	0.9224	0.6044	1.0000e-05
39/50	48ms	0.9501	0.5311	0.9224	0.6044	2.0000e-06
40/50	48ms	0.9538	0.5200	0.9225	0.6044	2.0000e-06
41/50	48ms	0.9557	0.5257	0.9224	0.6044	2.0000e-06
42/50	51ms	0.9539	0.5127	0.9225	0.6044	2.0000e-06

Berdasarkan training model 2 kelas, dapat disimpulkan bahwa model memiliki performa yang baik dalam klasifikasi gambar. Pada hasil akhir, model mencapai akurasi training sekitar 73,24% dan akurasi validasi sekitar 83,11%. Penggunaan algoritma pengoptimalan Adam dengan learning rate dinamis membantu meningkatkan performa model. Selama training, akurasi model mengalami peningkatan yang signifikan seiring dengan peningkatan jumlah epoch. Meskipun terdapat fluktuasi dalam validasi akurasi, secara keseluruhan terlihat tren peningkatan dengan stabil yang dapat dilihat pada tabel 4.25.

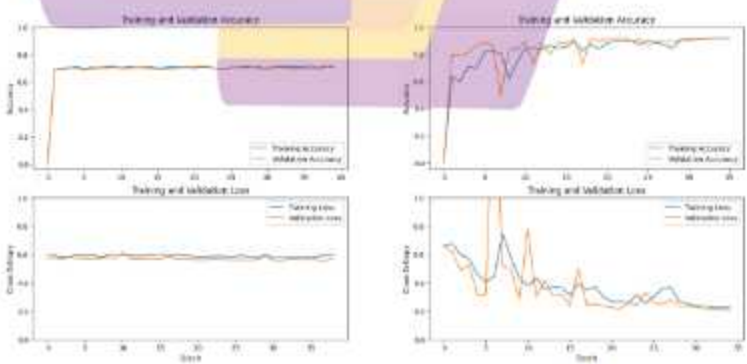
Tabel 4.25 Hasil ResNet50 50 Epoch 2 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/50	58ms	0.7717	0.4943	0.6876	0.5000	0.0010
2/50	49ms	0.7047	0.5376	0.6703	0.7022	0.0010
3/50	47ms	0.6834	0.5562	0.6555	0.7933	0.0010
4/50	43ms	0.6679	0.6024	0.6423	0.8400	0.0010
5/50	50ms	0.6580	0.6067	0.6321	0.7422	0.0010
6/50	50ms	0.6492	0.6305	0.6234	0.7844	0.0010
7/50	57ms	0.6443	0.6357	0.6185	0.6933	0.0010
8/50	48ms	0.6374	0.6581	0.6039	0.8400	0.0010
9/50	52ms	0.6200	0.6829	0.5959	0.8333	0.0010
10/50	44ms	0.6237	0.6762	0.6151	0.5867	0.0010
11/50	35ms	0.6169	0.6886	0.5929	0.7222	0.0010
12/50	35ms	0.6063	0.7076	0.6016	0.6333	0.0010
13/50	34ms	0.6144	0.6857	0.5739	0.8356	1.0000e-04
14/50	35ms	0.6088	0.6886	0.5743	0.8356	1.0000e-04
15/50	36ms	0.6036	0.6990	0.5746	0.8400	1.0000e-04
16/50	35ms	0.6008	0.7110	0.5740	0.8400	1.0000e-04
17/50	37ms	0.6031	0.7133	0.5728	0.8356	1.0000e-04
18/50	35ms	0.6056	0.7043	0.5722	0.8378	1.0000e-04
19/50	35ms	0.6020	0.7076	0.5725	0.8400	1.0000e-04
20/50	35ms	0.5929	0.7233	0.5709	0.8444	1.0000e-04
21/50	39ms	0.5955	0.7186	0.5707	0.8356	1.0000e-04
22/50	42ms	0.5995	0.7110	0.5710	0.8400	1.0000e-04
23/50	35ms	0.5996	0.7086	0.5695	0.8378	1.0000e-04
24/50	35ms	0.5962	0.7200	0.5700	0.8378	1.0000e-04
25/50	33ms	0.5957	0.7167	0.5704	0.8400	1.0000e-04
26/50	38ms	0.5948	0.7281	0.5675	0.8511	1.0000e-04
27/50	35ms	0.5909	0.7352	0.5666	0.8467	1.0000e-04
28/50	35ms	0.5952	0.7176	0.5661	0.8489	1.0000e-04
29/50	35ms	0.5954	0.7171	0.5659	0.8356	1.0000e-04
30/50	36ms	0.5950	0.7138	0.5664	0.8400	1.0000e-04
31/50	35ms	0.5916	0.7276	0.5643	0.8511	1.0000e-04
32/50	37ms	0.5953	0.7195	0.5655	0.8378	1.0000e-04
33/50	45ms	0.5897	0.7148	0.5646	0.8400	1.0000e-04
34/50	40ms	0.5835	0.7357	0.5631	0.8356	1.0000e-04

Tabel 4.25 Hasil ResNet50 50 Epoch 2 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
35/50	35ms	0.5889	0.7229	0.5632	0.8289	1.0000e-05
36/50	32ms	0.5873	0.7310	0.5632	0.8289	1.0000e-05
37/50	35ms	0.5885	0.7200	0.5630	0.8356	1.0000e-05
38/50	35ms	0.5889	0.7438	0.5630	0.8356	1.0000e-05
39/50	33ms	0.5941	0.7257	0.5632	0.8333	1.0000e-05
40/50	35ms	0.6017	0.7171	0.5629	0.8311	1.0000e-05
41/50	40ms	0.5856	0.7414	0.5628	0.8333	1.0000e-05
42/50	35ms	0.5933	0.7157	0.5629	0.8311	1.0000e-05
43/50	57ms	0.5921	0.7214	0.5629	0.8311	2.0000e-06
44/50	37ms	0.5945	0.7248	0.5629	0.8311	2.0000e-06
45/50	34ms	0.5914	0.7281	0.5629	0.8311	2.0000e-06
46/50	35ms	0.5894	0.7324	0.5629	0.8311	2.0000e-06

Kesimpulannya, meskipun terdapat fluktuasi dalam akurasi dan loss selama pelatihan, model berhasil mencapai akurasi yang relatif stabil pada akhirnya. Namun, peningkatan yang signifikan dalam akurasi validasi masih bisa menjadi perhatian untuk peningkatan lebih lanjut. Untuk itu, peneliti melakukan perbaikan menggunakan fine tuning yang berfungsi untuk melakukan penyesuaian learning rate yang diatur dari mulai terkecil sampai terbesar dengan menambahkan 30 epoch tambahan. Hasil dari fine tuning ini dapat dilihat pada Gambar 4.15.



(a) (b)

Gambar 4.15 Fine Tuning ResNet50 50 Epoch (CLAHE – White Balance). (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Dalam rangka meningkatkan performa model, peneliti melakukan fine tuning pada 3 kelas dengan menambahkan 30 epoch tambahan setelah epoch ke-50 yang dapat dilihat pada gambar 4.15 dan tabel 4.26. Hasil dari fine tuning tersebut menunjukkan bahwa loss pada data validasi tetap stabil di sekitar 56%-57%, sedangkan akurasi meningkat sekitar 70%-71%.

Tabel 4.26 Hasil Fine Tuning ResNet50 50 Epoch 3 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
42/80	77ms	0.6020	0.6946	0.5723	0.6948	2.0000e-06
43/80	62ms	0.5898	0.7003	0.5726	0.6874	2.0000e-06
44/80	65ms	0.5812	0.7013	0.5658	0.7007	2.0000e-06
45/80	63ms	0.5838	0.7076	0.5824	0.6963	2.0000e-06
46/80	63ms	0.5971	0.6962	0.5703	0.6859	2.0000e-06
47/80	62ms	0.5893	0.7041	0.5691	0.6919	2.0000e-06
48/80	65ms	0.5942	0.7000	0.5677	0.7052	2.0000e-06
49/80	63ms	0.5824	0.7095	0.5702	0.7022	2.0000e-06
50/80	63ms	0.5902	0.7124	0.5739	0.7052	2.0000e-06
51/80	62ms	0.6028	0.7006	0.5709	0.6993	2.0000e-06
52/80	63ms	0.5970	0.7076	0.6125	0.6948	2.0000e-06
53/80	65ms	0.5918	0.7048	0.5673	0.7067	2.0000e-06
54/80	63ms	0.5878	0.7038	0.5655	0.7067	2.0000e-06
55/80	63ms	0.5916	0.7070	0.5658	0.6919	2.0000e-06
56/80	63ms	0.5942	0.6997	0.5704	0.6948	2.0000e-06
57/80	63ms	0.5993	0.7022	0.5646	0.6889	2.0000e-06
58/80	63ms	0.5858	0.7092	0.5862	0.6874	2.0000e-06
59/80	63ms	0.5941	0.7048	0.5656	0.7052	2.0000e-06
60/80	63ms	0.5954	0.7092	0.5690	0.7037	2.0000e-06
61/80	66ms	0.5878	0.7010	0.5625	0.7096	2.0000e-06
62/80	66ms	0.5834	0.7079	0.5618	0.7126	2.0000e-06
63/80	62ms	0.5843	0.7065	0.5613	0.7096	2.0000e-06
64/80	64ms	0.5872	0.6997	0.5652	0.6963	2.0000e-06
65/80	63ms	0.5884	0.6940	0.5639	0.6889	2.0000e-06

Tabel 4.26 Hasil Fine Tuning ResNet50 50 Epoch 3 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
66/80	62ms	0.5845	0.7032	0.5616	0.7007	2.0000e-06
67/80	62ms	0.5858	0.7044	0.5602	0.7052	2.0000e-06
68/80	63ms	0.5907	0.7114	0.5609	0.7022	2.0000e-06
69/80	63ms	0.5818	0.7060	0.5587	0.7081	2.0000e-06
70/80	63ms	0.5780	0.7054	0.5661	0.6948	2.0000e-06
71/80	63ms	0.5992	0.6943	0.5816	0.7037	2.0000e-06
72/80	63ms	0.5763	0.7089	0.5597	0.7022	2.0000e-06
73/80	64ms	0.5787	0.7127	0.5569	0.7067	2.0000e-06
74/80	62ms	0.5797	0.7076	0.5644	0.6963	2.0000e-06
75/80	62ms	0.5797	0.7067	0.5681	0.6844	2.0000e-06
76/80	62ms	0.5815	0.7092	0.5605	0.7007	2.0000e-06
77/80	63ms	0.5793	0.7130	0.5734	0.6933	2.0000e-06
78/80	63ms	0.5864	0.6962	0.5577	0.7067	2.0000e-06
79/80	65ms	0.5936	0.7127	0.5575	0.7156	2.0000e-06
80/80	62ms	0.5953	0.7054	0.5741	0.7141	2.0000e-06

Sedangkan untuk 2 kelas, Hasil training tersebut menampilkan bahwa selama proses training, nilai loss dan akurasi dievaluasi untuk setiap epoch. Selama berjalannya epoch, terlihat adanya penurunan nilai loss dan peningkatan nilai akurasi secara bertahap. Pada epoch terakhir, diperoleh loss sebesar 58,94% dan akurasi sebesar 73,24% pada data training, sedangkan pada data validation diperoleh loss sebesar 56,29% dan akurasi sebesar 83,11% yang dapat dilihat pada gambar 4.15 dan tabel 4.27.

Tabel 4.27 Hasil Fine Tuning ResNet50 50 Epoch 2 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
46/80	68ms	0.6553	0.6343	0.6677	0.7978	1.0000e-04
47/80	59ms	0.6762	0.5976	0.6181	0.7889	1.0000e-04
48/80	55ms	0.5962	0.7076	0.4883	0.8111	1.0000e-04

Tabel 4.27 Hasil Fine Tuning ResNet50 50 Epoch 2 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
49/80	52ms	0.5730	0.6952	0.5289	0.8622	1.0000e-04
50/80	47ms	0.4602	0.8205	0.3167	0.8844	1.0000e-04
51/80	47ms	0.4057	0.8224	0.3126	0.8689	1.0000e-04
52/80	42ms	0.4457	0.8076	1.9581	0.5000	1.0000e-04
53/80	46ms	0.7426	0.6195	0.5186	0.8378	1.0000e-04
54/80	42ms	0.5733	0.7457	0.4989	0.8311	1.0000e-04
55/80	50ms	0.4325	0.8376	0.2855	0.8911	1.0000e-04
56/80	42ms	0.3787	0.8495	0.7718	0.7267	1.0000e-04
57/80	47ms	0.4365	0.8333	0.2953	0.8822	1.0000e-04
58/80	42ms	0.3559	0.8643	0.4139	0.8000	1.0000e-04
59/80	45ms	0.3679	0.8448	0.3072	0.8000	1.0000e-04
60/80	42ms	0.3655	0.8533	0.3187	0.8822	1.0000e-04
61/80	45ms	0.3138	0.8919	0.2391	0.9111	1.0000e-04
62/80	42ms	0.3922	0.8262	0.4951	0.7311	1.0000e-04
63/80	42ms	0.3476	0.8729	0.2435	0.9133	1.0000e-04
64/80	42ms	0.3712	0.8367	0.2510	0.8978	1.0000e-04
65/80	53ms	0.3005	0.8829	0.2373	0.9156	1.0000e-04
66/80	49ms	0.2663	0.8948	0.2243	0.9133	1.0000e-04
67/80	46ms	0.2677	0.9029	0.2188	0.9200	1.0000e-04
68/80	43ms	0.2627	0.8957	0.2552	0.8956	1.0000e-04
69/80	43ms	0.3163	0.8710	0.2394	0.9111	1.0000e-04
70/80	42ms	0.2555	0.9033	0.3334	0.8600	1.0000e-04
71/80	42ms	0.3004	0.8833	0.2622	0.8956	1.0000e-04
72/80	43ms	0.3580	0.8690	0.2464	0.8978	1.0000e-04
73/80	53ms	0.3687	0.8433	0.2834	0.8889	1.0000e-04
74/80	48ms	0.2689	0.9057	0.2340	0.9067	1.0000e-04
75/80	42ms	0.2514	0.9048	0.2341	0.9111	1.0000e-04
76/80	46ms	0.2353	0.9090	0.2303	0.9111	1.0000e-05
77/80	42ms	0.2377	0.9143	0.2188	0.9178	1.0000e-05
78/80	42ms	0.2240	0.9148	0.2164	0.9178	1.0000e-05
79/80	43ms	0.2254	0.9162	0.2086	0.9200	1.0000e-05
80/80	42ms	0.2248	0.9162	0.2340	0.9200	1.0000e-05

4.3.2. MobileNetV2

MobileNetV2 adalah arsitektur jaringan saraf konvolusi yang dikembangkan oleh Google (Sandler et al., 2018). Arsitektur ini dirancang khusus untuk aplikasi di perangkat mobile dan sumber daya terbatas, seperti ponsel pintar. Salah satu tujuan utama dari MobileNetV2 adalah menghasilkan model yang ringan dan efisien, sehingga cocok untuk diterapkan di perangkat dengan keterbatasan daya komputasi dan penyimpanan.

Dalam skenario kedua menggunakan arsitektur ResNet50 dengan dataset biasa, tahap awal yang dilakukan adalah preprocessing data. Preprocessing ini melibatkan pengolahan dan persiapan data agar siap digunakan sebagai input pada tahap pembelajaran.

```
def make_mobilenet_model(image_size, num_classes, data_augmentation =
data_augmenter()):
    input_shape = image_size + (3,)
    base_model = tf.keras.applications.MobileNetV2(input_shape=input_shape,
                                                    include_top=False,
                                                    weights="imagenet
    base_model.trainable = False
    inputs = tf.keras.Input(shape=input_shape)
    x = data_augmentation(inputs)
    x = base_model(x, training=False)
    x = GlobalAveragePooling2D()(x)
    x = Dropout(0.2)(x)
    if num_classes == 2:
        activation = "sigmoid"
        units = 2
    else:
        activation = "softmax"
        units = num_classes
    x = layers.Dropout(0.5)(x)

    prediction_layer = Dense(units, activation=activation)
    outputs = prediction_layer(x)
    model = Model(inputs, outputs)

    return model
```

Fungsi `make_mobilenet_model` digunakan untuk membuat model MobileNetV2. Fungsi ini menerima tiga parameter: `image_size` yang merupakan

dimensi gambar input, `num_classes` yang merupakan jumlah kelas output, dan `data_augmentation` yang merupakan fungsi augmentasi data opsional.

Pertama, fungsi ini menentukan bentuk input untuk model berdasarkan `image_size`, dengan menambahkan dimensi kedalaman 3. Selanjutnya, model dasar MobileNetV2 dibuat dengan menambahkan code pada program python `tf.keras.applications.MobileNetV2`. Lapisan teratas (klasifikasi) dari model tidak disertakan dengan mengatur `include_top=False`. Bobot model dasar diatur agar tidak dapat diperbarui dengan `base_model.trainable = False`. Selanjutnya, lapisan input dibuat menggunakan `tf.keras.Input` dengan bentuk yang telah ditentukan sebelumnya. Input yang telah di-augmentasi menggunakan fungsi `data_augmentation` diberikan ke model dasar MobileNetV2. Pada tahap ini, model dasar berada dalam mode inferensi (`training=False`). Kemudian, dilakukan operasi Global Average Pooling pada keluaran model dasar. Ini menghasilkan vektor fitur yang lebih kecil dengan mengambil rata-rata dari setiap fitur spasial.

Setelah itu, diterapkan lapisan Dropout dengan tingkat dropout sebesar 0.2 untuk mengurangi overfitting. Fungsi aktivasi dan jumlah unit untuk lapisan prediksi terakhir ditentukan berdasarkan `num_classes`. Jika `num_classes` adalah 2, fungsi aktivasi yang digunakan adalah "sigmoid" dengan 2 unit. Jika tidak, fungsi aktivasi yang digunakan adalah "softmax" dengan jumlah unit yang sama dengan `num_classes`. Selanjutnya, diterapkan lapisan Dropout tambahan dengan tingkat dropout sebesar 0.5. Ini juga bertujuan untuk mengurangi overfitting.

layer prediksi terakhir dibuat menggunakan `tf.keras.layers.Dense` dengan jumlah unit dan fungsi aktivasi yang telah ditentukan sebelumnya. Seluruh lapisan yang telah dibuat dihubungkan untuk membentuk model lengkap menggunakan `tf.keras.Model`. Selanjutnya pada peneliti menampulkan susunan layer dan membagi klasifikasi yang pada program sebagai berikut.

```
image_size = (224,224)
mobilenet_model = make_mobilenet_model(image_size, num_classes = 2)
mobilenet_model.summary()
```

program memanggil fungsi `make_mobilenet_model` dengan parameter `image_size` dan `num_classes`. Fungsi ini akan mengembalikan sebuah model MobileNet yang telah dikonfigurasi sesuai dengan parameter yang diberikan. Selanjutnya, program membuat objek mobilenet_model yang merupakan hasil dari pemanggilan fungsi `make_mobilenet_model`.

Terakhir, program mencetak ringkasan (summary) `mobilenet_model`, yang berisi informasi mengenai lapisan-lapisan dalam model beserta jumlah parameter yang digunakan. Dengan menjalankan program ini, kita dapat melihat struktur dan jumlah parameter dari model MobileNet yang telah dibuat, dari model ResNet50 yang telah dibuat yang dapat diliat pada tabel 4.4.

Table 4.28 Layer Model MobileNetV2

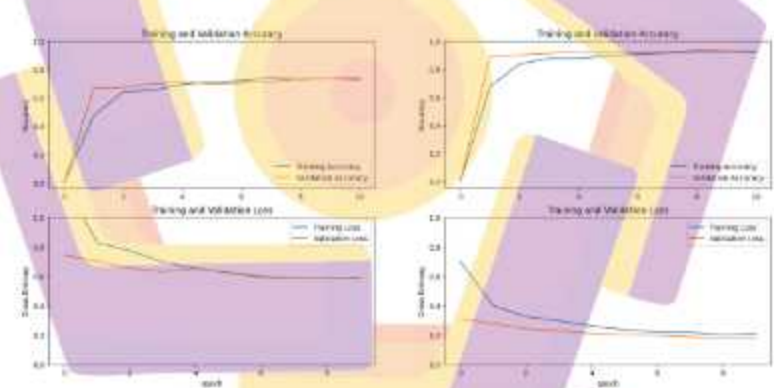
Layer (type)	**Output Shape**	**Param #**
input_6 (InputLayer)	[(None, 224, 224, 3)]	0
sequential_2 (Sequential)	(None, 224, 224, 3)	0
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2257984
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
Dropout_4 (Dropout)	(None, 1280)	0
dropout_5 (Dropout)	(None, 1280)	0
Dense_2 (Dense)	(None, 2)	2562

Total params: 2,260,546
Trainable params: 2,562
Non-trainable params: 2,257,984

Dalam pelatihan data, terdapat dua skenario yang digunakan. Skenario pertama melibatkan tiga kelas, yaitu Normal CXR, Bakteri Pneumonia CXR, dan Virus Pneumonia CXR. Skenario kedua melibatkan dua kelas, yaitu Normal CXR dan Pneumonia CXR.

4.3.2.1. Training Validation 10 Epoch

Berikut adalah hasil akurasi dan loss dari pelatihan dan validasi data dalam Gambar 4.16 untuk dua skenario, yaitu (a) dengan 3 kelas dan (b) dengan 2 kelas.



(a) (b)

Gambar 4.16 MobileNetV2 10 Epoch. (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.16 dan tabel 4.29 dapat terlihat hasil klasifikasi 3 kelas, bahwa selama proses training dilakukan evaluasi terhadap loss dan akurasi model untuk setiap epoch. Dari data tersebut, terlihat bahwa nilai loss cenderung menurun dan akurasi cenderung meningkat seiring dengan peningkatan jumlah epoch. Pada

epoch 10, diperoleh loss sebesar 58,85% dan akurasi sebesar 72,59% pada data validasi. Hal ini menunjukkan bahwa model telah mengalami peningkatan performa sejak awal training. Selain itu, terlihat bahwa learning rate tetap pada nilai 0,001 selama keseluruhan proses pelatihan.

Tabel 4.29 Hasil MobileNetV2 10 Epoch 3 Kelas

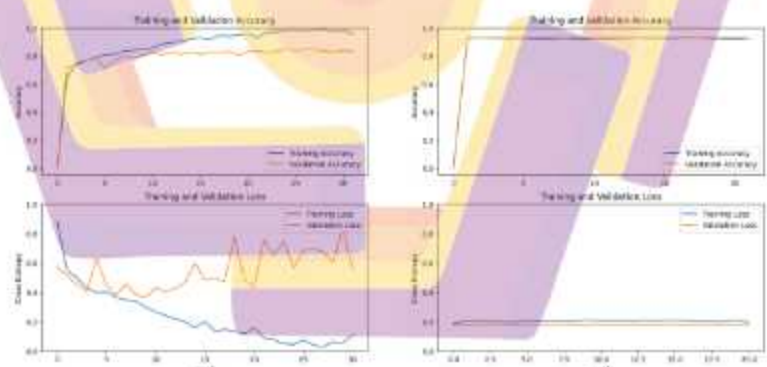
Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/10	47ms	1.2371	0.4743	0.7414	0.6667	0.0010
2/10	23ms	0.8289	0.6438	0.7035	0.6741	0.0010
3/10	23ms	0.7740	0.6524	0.6541	0.6978	0.0010
4/10	23ms	0.6967	0.6962	0.6311	0.7111	0.0010
5/10	22ms	0.6537	0.7114	0.6515	0.6978	0.0010
6/10	21ms	0.6233	0.7219	0.6263	0.7037	0.0010
7/10	23ms	0.5940	0.7406	0.5981	0.7170	0.0010
8/10	23ms	0.5900	0.7298	0.5916	0.7333	0.0010
9/10	24ms	0.5861	0.7368	0.5856	0.7363	0.0010
10/10	22ms	0.5831	0.7375	0.5885	0.7259	0.0010

Sedangkan pada 2 kelas dapat dilihat dalam gambar tersebut,bahwa selama proses training dilakukan evaluasi terhadap loss dan akurasi model untuk setiap epoch. Dari data tersebut, terlihat bahwa nilai loss cenderung menurun dan akurasi cenderung meningkat seiring dengan peningkatan jumlah epoch. Pada epoch 10, diperoleh loss sebesar 17,59% dan akurasi sebesar 92,89% pada data validasi. Hal ini menunjukkan bahwa model telah mengalami peningkatan performa sejak awal training. Selain itu, terlihat bahwa learning rate tetap pada nilai 0,001 selama keseluruhan proses pelatihan. Model mencapai hasil akurasi yang tinggi, yaitu sekitar 92,89%, dan loss yang rendah, yaitu sekitar 17,59% yang dapat dilihat pada tabel 4.30.

Tabel 4.30 Hasil MobileNetV2 10 Epoch 2 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/10	59ms	0.7066	0.6767	0.3105	0.8889	0.0010
2/10	23ms	0.3998	0.8371	0.2780	0.9022	0.0010
3/10	23ms	0.3243	0.8781	0.2425	0.9133	0.0010
4/10	23ms	0.2969	0.8771	0.2286	0.9244	0.0010
5/10	21ms	0.2622	0.8976	0.2109	0.9222	0.0010
6/10	21ms	0.2353	0.9076	0.2012	0.9222	0.0010
7/10	21ms	0.2203	0.9157	0.1979	0.9200	0.0010
8/10	24ms	0.2177	0.9176	0.1879	0.9356	0.0010
9/10	22ms	0.2007	0.9281	0.1783	0.9311	0.0010
10/10	22ms	0.2090	0.9190	0.1759	0.9289	0.0010

Setelah itu, peneliti mengoptimalkan model dengan melakukan perbaikan menggunakan fine tuning yang berfungsi untuk melakukan penyesuaian learning rate yang diatur dari mulai terkecil sampai terbesar dengan menambahkan 30 epoch tambahan. Perubahan hasil fine tuning ini dapat ditemukan di Gambar 4.17.



(a) (b)

Gambar 4.17 Fine Tuning MobileNetV2 10 Epoch. (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Setelah melakukan tahap finetuning pada 3 kelas, proses epoch terakhir diperoleh loss sebesar 55,57% dan akurasi sebesar 82,96% pada data validasi sedangkan untuk data training diperoleh loss 10,90% dan akurasi 95,90%. Dalam

proses pelatihan, terlihat bahwa model mengalami peningkatan performa pada awal-awal epoch. Namun, setelah beberapa epoch, terjadi fluktuasi dalam performa model dengan loss dan akurasi yang naik-turun secara tidak konsisten yang dapat dilihat pada gambar 4.17 dan tabel 4.31.

Tabel 4.31 Hasil Fine Tuning MobileNetV2 10 Epoch 3 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
10/40	105ms	0.8868	0.6327	0.5701	0.7170	1.0000e-04
11/40	67ms	0.5458	0.7479	0.5112	0.7333	1.0000e-04
12/40	68ms	0.4989	0.7625	0.4550	0.7630	1.0000e-04
13/40	68ms	0.4281	0.7867	0.4043	0.7881	1.0000e-04
14/40	66ms	0.3980	0.8076	0.6299	0.7126	1.0000e-04
15/40	66ms	0.4018	0.8098	0.5566	0.7644	1.0000e-04
16/40	68ms	0.3629	0.8343	0.3803	0.8044	1.0000e-04
17/40	66ms	0.3467	0.8390	0.5430	0.7822	1.0000e-04
18/40	66ms	0.3383	0.8451	0.3808	0.8044	1.0000e-04
19/40	67ms	0.2973	0.8562	0.3649	0.8207	1.0000e-04
20/40	66ms	0.2658	0.8841	0.4300	0.8089	1.0000e-04
21/40	67ms	0.2420	0.8924	0.4033	0.8237	1.0000e-04
22/40	66ms	0.2204	0.9013	0.4240	0.8104	1.0000e-04
23/40	67ms	0.1973	0.9190	0.4684	0.8267	1.0000e-04
24/40	66ms	0.1577	0.9292	0.5975	0.8089	1.0000e-04
25/40	65ms	0.2014	0.9149	0.4830	0.8237	1.0000e-04
26/40	65ms	0.1338	0.9441	0.4907	0.8252	1.0000e-04
27/40	67ms	0.1462	0.9406	0.4734	0.8311	1.0000e-04
28/40	66ms	0.1344	0.9460	0.7790	0.8059	1.0000e-04
29/40	66ms	0.1146	0.9559	0.4860	0.8296	1.0000e-04
30/40	67ms	0.4571	0.9311	0.4353	0.8341	1.0000e-04
31/40	66ms	0.0879	0.9673	0.7569	0.8267	1.0000e-04
32/40	66ms	0.0758	0.9705	0.6468	0.8296	1.0000e-04
33/40	66ms	0.0493	0.9810	0.7457	0.8474	1.0000e-04
34/40	66ms	0.0426	0.9857	0.5640	0.8370	1.0000e-04
35/40	66ms	0.0759	0.9762	0.6834	0.8444	1.0000e-04
36/40	66ms	0.0400	0.9851	0.6964	0.8430	1.0000e-04
37/40	66ms	0.0281	0.9892	0.6739	0.8356	1.0000e-04
38/40	66ms	0.0624	0.9740	0.6023	0.8252	1.0000e-04

Tabel 4.31 Hasil Fine Tuning MobileNetV2 10 Epoch 3 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
39/40	66ms	0.0570	0.9775	0.8437	0.8400	1.0000e-04
40/40	65ms	0.1090	0.9590	0.5557	0.8296	1.0000e-04

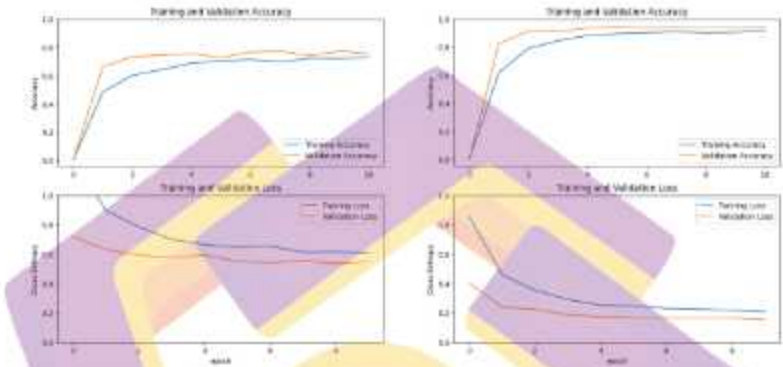
Sedangkan untuk 2 kelas, dapat disimpulkan bahwa model hasil training memiliki akurasi yang stabil pada tingkat sekitar 92,89% pada data validasi setelah 30 epoch. Model juga menunjukkan peningkatan dalam loss pada awal training, namun kemudian tidak ada perubahan signifikan setelah beberapa epoch yang dapat dilihat pada tabel 4.32.

Tabel 4.32 Hasil Fine Tuning MobileNetV2 10 Epoch 2 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
10/40	55ms	0.1891	0.9243	0.1759	0.9289	1.0000e-04
11/40	21ms	0.2056	0.9281	0.1759	0.9289	1.0000e-04
12/40	21ms	0.2040	0.9252	0.1759	0.9289	1.0000e-04
13/40	21ms	0.1989	0.9243	0.1759	0.9289	1.0000e-04
14/40	21ms	0.1964	0.9257	0.1759	0.9289	1.0000e-04
15/40	21ms	0.2037	0.9190	0.1759	0.9289	1.0000e-04
16/40	21ms	0.2028	0.9214	0.1759	0.9289	1.0000e-04
17/40	21ms	0.2033	0.9252	0.1759	0.9289	1.0000e-04
18/40	21ms	0.2004	0.9157	0.1759	0.9289	1.0000e-04
19/40	21ms	0.2115	0.9233	0.1759	0.9289	1.0000e-05
20/40	21ms	0.2049	0.9214	0.1759	0.9289	1.0000e-05
21/40	21ms	0.2069	0.9248	0.1759	0.9289	1.0000e-05
22/40	21ms	0.2019	0.9281	0.1759	0.9289	1.0000e-05
23/40	22ms	0.2091	0.9257	0.1759	0.9289	1.0000e-05
24/40	21ms	0.2050	0.9252	0.1759	0.9289	1.0000e-05
25/40	21ms	0.2099	0.9252	0.1759	0.9289	1.0000e-05
26/40	21ms	0.2043	0.9305	0.1759	0.9289	1.0000e-05
27/40	21ms	0.2014	0.9248	0.1759	0.9289	2.0000e-06
28/40	21ms	0.1973	0.9219	0.1759	0.9289	2.0000e-06
29/40	21ms	0.2102	0.9181	0.1759	0.9289	2.0000e-06
30/40	24ms	0.1943	0.9248	0.1759	0.9289	2.0000e-06

4.3.2.2. Training Validation 10 Epoch (CLAHE – White Balance)

Berikut adalah hasil akurasi dan loss dari pelatihan dan validasi data dalam Gambar 4.18 untuk dua skenario, yaitu (a) dengan 3 kelas dan (b) dengan 2 kelas.



(a) (b)

Gambar 4.18 MobileNetV2 10 Epoch (CLAHE – White Balance). (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.8 dan tabel 4.33 dapat terlihat hasil klasifikasi 3 kelas dengan 10 epoch yang dilakukan, terdapat perhitungan loss dan akurasi pada data training dan data validation. Pada epoch pertama, didapatkan loss sebesar 125,04% dan akurasi sebesar 48,32% pada data training, serta loss sebesar 71,93% dan akurasi sebesar 66,22% pada data validasi. Pada epoch selanjutnya, terjadi peningkatan akurasi dan penurunan loss pada kedua dataset. Pada akhir training dengan epoch ke-10, didapatkan loss sebesar 61,02% dan akurasi sebesar 72,70% pada data training, serta loss sebesar 54,79% dan akurasi sebesar 75,41% pada data validasi.

Tabel 4.33 Hasil MobileNetV2 10 Epoch 3 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/10	66ms	1.2504	0.4832	0.7193	0.6622	0.0010
2/10	33ms	0.8966	0.6013	0.6307	0.7333	0.0010
3/10	34ms	0.7848	0.6425	0.5927	0.7437	0.0010

Tabel 4.33 Hasil MobileNetV2 10 Epoch 3 Kelas (CLAHE – White Balance) (Lanjutan)

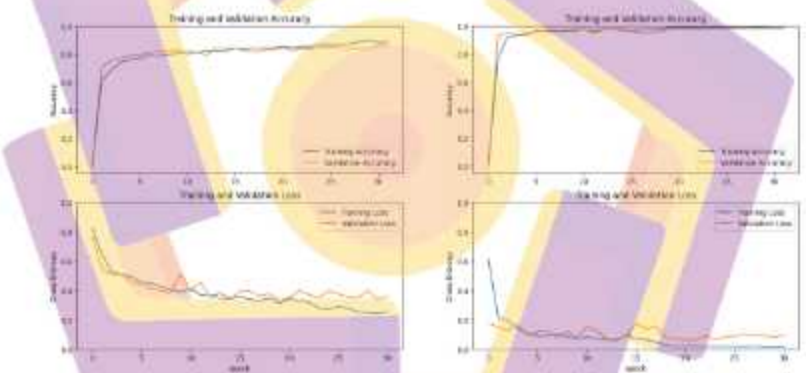
Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
4/10	33ms	0.7000	0.6879	0.5760	0.7570	0.0010
5/10	31ms	0.6632	0.7025	0.5941	0.7259	0.0010
6/10	33ms	0.6480	0.7140	0.5497	0.7659	0.0010
7/10	33ms	0.6539	0.6959	0.5414	0.7719	0.0010
8/10	32ms	0.6130	0.7225	0.5540	0.7407	0.0010
9/10	32ms	0.6151	0.7200	0.5344	0.7733	0.0010
10/10	32ms	0.6102	0.7270	0.5479	0.7541	0.0010

Sedangan untuk 2 kelas pada setiap epoch, dilakukan perhitungan loss dan akurasi pada data training dan data validasi. Pada epoch pertama, didapatkan loss sebesar 85,38% dan akurasi sebesar 61,81% pada data training, serta loss sebesar 39,95% dan akurasi sebesar 82,44% pada data validasi. Pada epoch-epoch selanjutnya, terjadi peningkatan akurasi dan penurunan loss pada kedua dataset. Pada akhir pelatihan dengan epoch ke-10, didapatkan loss sebesar 20,67% dan akurasi sebesar 92,00% pada data pelatihan, serta loss sebesar 15,41% dan akurasi sebesar 94,22% pada data validasi dapat dilihat pada tabel 4.34.

Tabel 4.34 Hasil MobileNetV2 10 Epoch 2 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/10	78ms	0.8538	0.6181	0.3995	0.8244	0.0010
2/10	34ms	0.4609	0.7905	0.2382	0.9156	0.0010
3/10	32ms	0.3540	0.8452	0.2254	0.9089	0.0010
4/10	34ms	0.2916	0.8824	0.1863	0.9356	0.0010
5/10	34ms	0.2512	0.8952	0.1738	0.9444	0.0010
6/10	31ms	0.2481	0.9029	0.1675	0.9356	0.0010
7/10	32ms	0.2298	0.9095	0.1638	0.9422	0.0010
8/10	31ms	0.2264	0.9019	0.1666	0.9356	0.0010
9/10	31ms	0.2165	0.9048	0.1636	0.9378	0.0010
10/10	31ms	0.2067	0.9200	0.1541	0.9422	0.0010

Dari hasil tersebut, dapat disimpulkan bahwa model mengalami peningkatan performa seiring berjalannya epoch. Akurasi pada data training dan data validasi cenderung meningkat, sedangkan loss cenderung menurun. Hal ini menunjukkan bahwa model dapat belajar dari data training dan mampu melakukan klasifikasi dengan baik pada data validasi. Setelah itu, peneliti mengoptimalkan model dengan melakukan perbaikan menggunakan fine tuning yang berfungsi untuk melakukan penyesuaian learning rate yang diatur dari mulai terkecil sampai terbesar dengan menambahkan 30 epoch tambahan. Perubahan hasil finetuning ini dapat ditemukan di Gambar 4.19.



(a) (b)

Gambar 4.19 Fine Tuning MobileNetV2 10 Epoch (CLAHE – White Balance). (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.19 dan tabel 3.35 dapat dilihat setelah melalui tahap finetuning pada 3 kelas model mencapai akurasi sekitar 88,76% pada data training dan 85,78% pada data validasi. Model tersebut memiliki loss sebesar 25,61% pada data training dan 36,42% pada data validasi.

Tabel 4.35 Hasil Fine Tuning MobileNetV2 10 Epoch 3 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
10/40	122ms	0.8258	0.6076	0.7733	0.7081	1.0000e-04
11/40	77ms	0.6488	0.6873	0.5440	0.7556	1.0000e-04
12/40	77ms	0.5277	0.7429	0.5122	0.7704	1.0000e-04
13/40	76ms	0.5070	0.7644	0.5069	0.7911	1.0000e-04
14/40	77ms	0.4953	0.7695	0.4590	0.7970	1.0000e-04
15/40	76ms	0.4578	0.7940	0.4395	0.8104	1.0000e-04
16/40	77ms	0.4493	0.7854	0.4173	0.8207	1.0000e-04
17/40	75ms	0.4273	0.7994	0.3973	0.8193	1.0000e-04
18/40	76ms	0.4061	0.8051	0.3885	0.8341	1.0000e-04
19/40	75ms	0.3897	0.8206	0.5140	0.7985	1.0000e-04
20/40	76ms	0.4138	0.8105	0.3975	0.8169	1.0000e-04
21/40	75ms	0.3707	0.8279	0.4565	0.7956	1.0000e-04
22/40	77ms	0.3628	0.8241	0.3593	0.8459	1.0000e-04
23/40	76ms	0.3802	0.8238	0.3468	0.8415	1.0000e-04
24/40	75ms	0.3452	0.8454	0.3410	0.8459	1.0000e-04
25/40	75ms	0.3582	0.8349	0.3942	0.8222	1.0000e-04
26/40	75ms	0.3403	0.8441	0.3951	0.8148	1.0000e-04
27/40	75ms	0.3315	0.8463	0.3627	0.8311	1.0000e-04
28/40	74ms	0.3503	0.8454	0.3798	0.8326	1.0000e-04
29/40	77ms	0.3126	0.8562	0.3320	0.8489	1.0000e-04
30/40	76ms	0.3454	0.8454	0.3497	0.8296	1.0000e-04
31/40	74ms	0.3303	0.8562	0.3994	0.8370	1.0000e-04
32/40	75ms	0.3194	0.8546	0.3735	0.8341	1.0000e-04
33/40	76ms	0.2814	0.8689	0.3504	0.8504	1.0000e-04
34/40	76ms	0.2736	0.8746	0.4002	0.8563	1.0000e-04
35/40	75ms	0.2924	0.8632	0.3766	0.8430	1.0000e-04
36/40	76ms	0.2774	0.8746	0.3546	0.8415	1.0000e-04
37/40	75ms	0.2563	0.8889	0.3521	0.8444	1.0000e-04
38/40	75ms	0.2488	0.8956	0.4013	0.8385	1.0000e-04
39/40	77ms	0.2430	0.8841	0.3417	0.8607	1.0000e-04
40/40	75ms	0.2561	0.8876	0.3642	0.8578	1.0000e-04

Sedangkan pada 2 kelas, Pada epoch ke-1, model mencapai akurasi pelatihan sebesar 42,38% dan akurasi validasi sebesar 51,56%. Selama pelatihan,

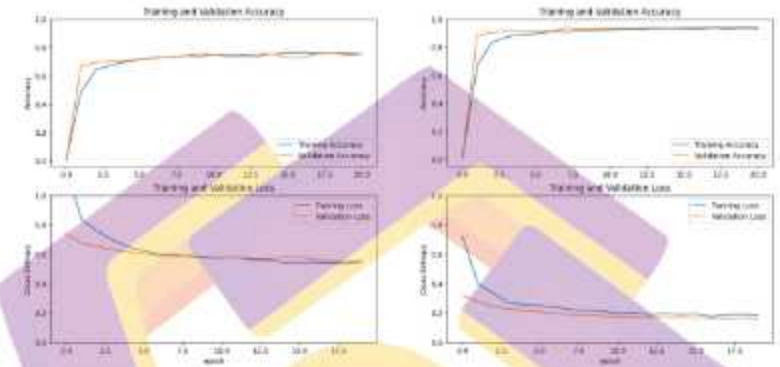
akurasi model meningkat secara bertahap dan mencapai akurasi pelatihan sebesar 99,33% dan akurasi validasi sebesar 97,78% pada hasil akhir yang dapat dilihat pada tabel 4.36.

Tabel 4.36 Hasil Fine Tuning MobileNetV2 10 Epoch 2 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
10/40	147ms	0.6165	0.7657	0.1781	0.9356	1.0000e-04
11/40	78ms	0.2144	0.9224	0.1508	0.9489	1.0000e-04
12/40	76ms	0.1962	0.9348	0.1307	0.9400	1.0000e-04
13/40	76ms	0.1400	0.9476	0.1645	0.9333	1.0000e-04
14/40	77ms	0.1085	0.9614	0.0996	0.9689	1.0000e-04
15/40	77ms	0.0891	0.9681	0.1148	0.9622	1.0000e-04
16/40	76ms	0.0922	0.9686	0.1323	0.9578	1.0000e-04
17/40	77ms	0.1069	0.9605	0.0794	0.9756	1.0000e-04
18/40	76ms	0.0853	0.9738	0.1236	0.9600	1.0000e-04
19/40	78ms	0.0734	0.9743	0.0783	0.9800	1.0000e-04
20/40	76ms	0.0836	0.9676	0.1513	0.9533	1.0000e-04
21/40	76ms	0.0746	0.9738	0.1297	0.9711	1.0000e-04
22/40	75ms	0.0650	0.9786	0.0672	0.9778	1.0000e-04
23/40	76ms	0.0650	0.9767	0.0604	0.9689	1.0000e-04
24/40	76ms	0.0943	0.9633	0.0928	0.9667	1.0000e-04
25/40	76ms	0.0732	0.9724	0.1756	0.9511	1.0000e-04
26/40	75ms	0.0659	0.9733	0.1387	0.9600	1.0000e-04
27/40	75ms	0.0489	0.9870	0.1573	0.9600	1.0000e-04
28/40	75ms	0.0299	0.9895	0.0807	0.9800	1.0000e-05
29/40	76ms	0.0210	0.9938	0.0741	0.9800	1.0000e-05
30/40	76ms	0.0254	0.9900	0.0622	0.9800	1.0000e-05
31/40	78ms	0.0241	0.9900	0.0637	0.9822	1.0000e-05
32/40	75ms	0.0223	0.9895	0.0923	0.9733	1.0000e-05
33/40	76ms	0.0197	0.9919	0.0675	0.9800	1.0000e-05
34/40	75ms	0.0209	0.9900	0.0869	0.9778	1.0000e-05
35/40	76ms	0.0192	0.9929	0.0891	0.9822	1.0000e-05
36/40	76ms	0.0172	0.9905	0.1051	0.9778	1.0000e-05
37/40	76ms	0.0215	0.9905	0.0944	0.9778	1.0000e-05
38/40	75ms	0.0159	0.9924	0.0928	0.9778	1.0000e-05
39/40	75ms	0.0152	0.9957	0.0830	0.9822	1.0000e-05
40/40	76ms	0.0140	0.9933	0.0974	0.9778	2.0000e-06

4.3.2.3. Training Validation 20 Epoch

Berikut adalah hasil akurasi dan loss dari pelatihan dan validasi data dalam Gambar 4.20 untuk dua skenario, yaitu (a) dengan 3 kelas dan (b) dengan 2 kelas.



(a) (b)

Gambar 4.20 MobileNetV2 20 Epoch. (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.20 dan tabel 4.37 dapat dilihat skema klasifikasi 3 kelas pada epoch pertama, akurasi model hanya sekitar 48,63% dengan loss sebesar 118,11%. Namun, akurasi secara bertahap meningkat seiring dengan training model dan pada epoch 20 akurasi mencapai 75,52% dengan loss 54,60%. Selain itu, pada data validasi, model juga menunjukkan peningkatan yang cukup signifikan. Pada awalnya, akurasi validasi sekitar 66,81%, dan pada epoch terakhir, akurasi mencapai 75,11%. Loss pada data validasi juga menurun dari 73,97% menjadi 55,63%.

Tabel 4.37 Hasil MobileNetV2 20 Epoch 3 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/20	47ms	1.1811	0.4863	0.7397	0.6681	0.0010
2/20	22ms	0.8312	0.6397	0.6736	0.6919	0.0010
3/20	22ms	0.7574	0.6711	0.6505	0.7052	0.0010

Tabel 4.37 Hasil MobileNetV2 20 Epoch 3 Kelas (Lanjutan)

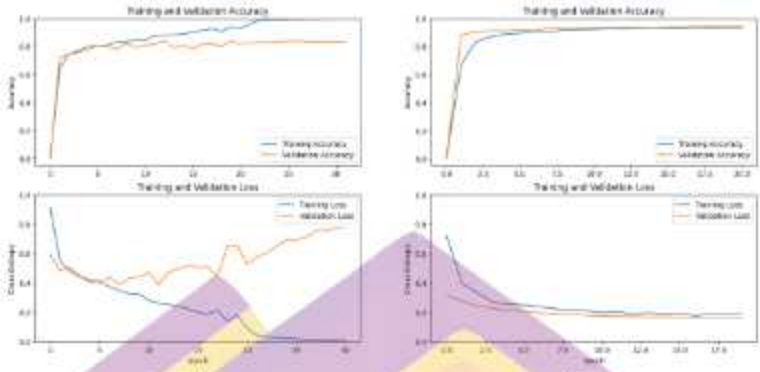
Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
4/20	22ms	0.6942	0.6956	0.6294	0.7067	0.0010
5/20	23ms	0.6472	0.7143	0.6142	0.7156	0.0010
6/20	22ms	0.6220	0.7267	0.6005	0.7215	0.0010
7/20	22ms	0.5984	0.7311	0.5882	0.7319	0.0010
8/20	22ms	0.5952	0.7384	0.5871	0.7393	0.0010
9/20	21ms	0.5837	0.7502	0.5903	0.7348	0.0010
10/20	23ms	0.5737	0.7454	0.5793	0.7467	0.0010
11/20	22ms	0.5717	0.7476	0.5728	0.7348	0.0010
12/20	22ms	0.5768	0.7438	0.5756	0.7289	0.0010
13/20	22ms	0.5600	0.7498	0.5679	0.7333	0.0010
14/20	23ms	0.5586	0.7524	0.5666	0.7541	0.0010
15/20	22ms	0.5410	0.7613	0.5772	0.7259	0.0010
16/20	22ms	0.5429	0.7641	0.5803	0.7255	0.0010
17/20	21ms	0.5426	0.7587	0.5621	0.7526	0.0010
18/20	23ms	0.5375	0.7597	0.5575	0.7600	0.0010
19/20	22ms	0.5348	0.7587	0.5577	0.7422	0.0010
20/20	22ms	0.5460	0.7552	0.5563	0.7511	0.0010

Selanjutnya pada gambar tersebut menampilkan hasil dari skema klasifikasi 2 kelas, dapat disimpulkan bahwa model yang dilatih memiliki peningkatan akurasi dari epoch ke epoch. Pada awalnya, akurasi model sekitar 67,05% dengan loss sebesar 72,47%. Namun, akurasi secara bertahap meningkat seiring dengan training model dan pada epoch terakhir akurasi mencapai 93,19% dengan loss 18,43%. Pada data validasi, model juga menunjukkan peningkatan yang signifikan. Pada awalnya, akurasi validasi sekitar 88,44%, dan pada epoch 20 akurasi mencapai 94,22%. LSedangkan loss pada data validasi juga menurun dari 31,89% menjadi 15,88% yang dapat dilihat pada gambar 4.20 dan tabel 4.38.

Tabel 4.38 Hasil MobileNetV2 20 Epoch 2 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/20	60ms	0.7247	0.6705	0.3189	0.8844	0.0010
2/20	24ms	0.3977	0.8329	0.2763	0.9022	0.0010
3/20	23ms	0.3324	0.8710	0.2419	0.9133	0.0010
4/20	22ms	0.2690	0.8900	0.2282	0.9111	0.0010
5/20	24ms	0.2535	0.8948	0.2139	0.9200	0.0010
6/20	24ms	0.2499	0.9095	0.2062	0.9133	0.0010
7/20	24ms	0.2372	0.9062	0.1949	0.9356	0.0010
8/20	22ms	0.2231	0.9181	0.1887	0.9267	0.0010
9/20	22ms	0.2150	0.9210	0.1864	0.9267	0.0010
10/20	22ms	0.2102	0.9229	0.1784	0.9333	0.0010
11/20	22ms	0.1965	0.9281	0.1751	0.9333	0.0010
12/20	22ms	0.2037	0.9262	0.1710	0.9333	0.0010
13/20	24ms	0.1875	0.9319	0.1700	0.9378	0.0010
14/20	22ms	0.1958	0.9324	0.1672	0.9356	0.0010
15/20	24ms	0.1888	0.9319	0.1691	0.9400	0.0010
16/20	22ms	0.1923	0.9295	0.1652	0.9400	0.0010
17/20	24ms	0.1757	0.9352	0.1611	0.9422	0.0010
18/20	22ms	0.1872	0.9305	0.1594	0.9422	0.0010
19/20	21ms	0.1888	0.9338	0.1639	0.9422	0.0010
20/20	21ms	0.1843	0.9319	0.1588	0.9422	0.0010

Setelah itu, dilakukan perbaikan menggunakan fine tuning yang berfungsi untuk melakukan penyesuaian learning rate yang diatur dari mulai terkecil sampai terbesar dengan menambahkan 30 epoch tambahan untuk mengoptimalkan model. Perubahan hasil fine tuning ini dapat ditemukan di Gambar 4.21.



(a) (b)

Gambar 4.21 Fine Tuning MobileNetV2 20 Epoch. (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.21 dan tabel 4.39 dapat dilihat bahwa klasifikasi 3 kelas memiliki hasil training yang menunjukan penurunan loss dan peningkatan akurasi seiring berjalannya epoch. Pada akhir training, model mencapai loss sebesar 0,83% dan akurasi sebesar 99,87% pada data pelatihan. Pada data validasi, model mencapai loss sebesar 77,78% dan akurasi sebesar 83,41%.

Tabel 4.39 Hasil Fine Tuning MobileNetV2 20 Epoch 3 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
20/50	104ms	0.9120	0.6451	0.5924	0.7185	1.0000e-04
21/50	67ms	0.5485	0.7463	0.4823	0.7467	1.0000e-04
22/50	66ms	0.4804	0.7705	0.5052	0.7570	1.0000e-04
23/50	66ms	0.4405	0.7994	0.4467	0.7808	1.0000e-04
24/50	67ms	0.4222	0.8041	0.4069	0.8030	1.0000e-04
25/50	65ms	0.4125	0.8083	0.3985	0.8030	1.0000e-04
26/50	65ms	0.3755	0.8311	0.4384	0.7793	1.0000e-04
27/50	67ms	0.3474	0.8343	0.3881	0.8207	1.0000e-04
28/50	65ms	0.3284	0.8460	0.4302	0.7911	1.0000e-04
29/50	65ms	0.3250	0.6454	0.4395	0.8089	1.0000e-04
30/50	65ms	0.2814	0.8775	0.4717	0.8148	1.0000e-04
31/50	67ms	0.2608	0.8800	0.3883	0.8430	1.0000e-04
32/50	66ms	0.2526	0.8838	0.4743	0.7911	1.0000e-04

Tabel 4.39 Hasil Fine Tuning MobileNetV2 20 Epoch 3 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
33/50	65ms	0.2421	0.8895	0.4907	0.8074	1.0000e-04
34/50	65ms	0.2235	0.9025	0.5190	0.7852	1.0000e-04
35/50	65ms	0.1991	0.9156	0.5061	0.8193	1.0000e-04
36/50	65ms	0.1889	0.9260	0.5087	0.8148	1.0000e-04
37/50	65ms	0.2146	0.9067	0.4428	0.8030	1.0000e-04
38/50	65ms	0.1387	0.9413	0.6533	0.8385	1.0000e-04
39/50	65ms	0.1801	0.9267	0.6524	0.8193	1.0000e-04
40/50	65ms	0.0888	0.9654	0.5256	0.8281	1.0000e-05
41/50	65ms	0.0430	0.9851	0.5761	0.8296	1.0000e-05
42/50	65ms	0.0301	0.9923	0.6058	0.8296	1.0000e-05
43/50	65ms	0.0263	0.9917	0.6458	0.8356	1.0000e-05
44/50	65ms	0.0224	0.9940	0.6947	0.8370	1.0000e-05
45/50	65ms	0.0200	0.9946	0.6919	0.8400	1.0000e-05
46/50	65ms	0.0131	0.9975	0.7128	0.8341	1.0000e-05
47/50	65ms	0.0121	0.9978	0.7575	0.8341	1.0000e-05
48/50	66ms	0.0095	0.9987	0.7548	0.8326	2.0000e-06
49/50	65ms	0.0088	0.9978	0.7717	0.8341	2.0000e-06
50/50	65ms	0.0083	0.9987	0.7778	0.8341	2.0000e-06

Sedangkan untuk 2 kelas, hasil dari klasifikasi menunjukan bahwa memiliki akurasi sekitar 67,05% dengan loss 72,47%. Namun, melalui training selama 20 epoch, akurasi secara bertahap meningkat hingga mencapai 93,19% dengan loss 18,43% pada 20 epoch. Pada data validasi, model juga menunjukan peningkatan yang signifikan. Awalnya, akurasi validasi sekitar 88,44% dan pada epoch 20 mencapai akurasi 94,22% dengan loss 15,88% yang dapat dilihat pada tabel 4.40.

Tabel 4.40 Hasil Fine Tuning MobileNetV2 20 Epoch 2 Kelas

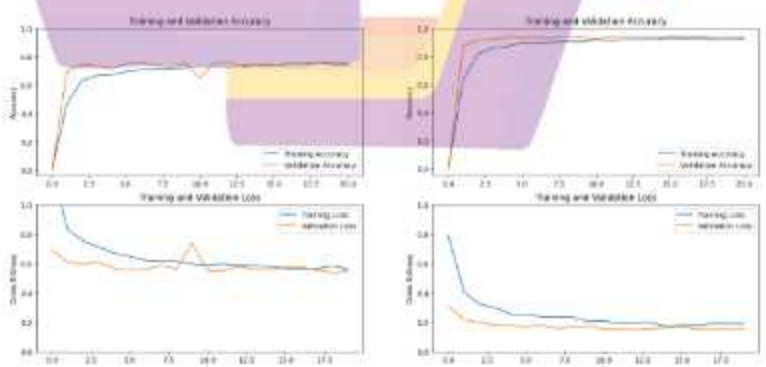
Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
20/50	56ms	0.1781	0.9338	0.1588	0.9422	1.0000e-04
21/50	22ms	0.1723	0.9419	0.1588	0.9422	1.0000e-04
22/50	22ms	0.1822	0.9381	0.1588	0.9422	1.0000e-04
23/50	22ms	0.1892	0.9348	0.1588	0.9422	1.0000e-04

Tabel 4.40 Hasil Fine Tuning MobileNetV2 20 Epoch 2 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
24/50	22ms	0.1856	0.9362	0.1588	0.9422	1.0000e-04
25/50	22ms	0.1859	0.9371	0.1588	0.9422	1.0000e-04
26/50	22ms	0.1783	0.9386	0.1588	0.9422	1.0000e-04
27/50	22ms	0.1864	0.9329	0.1588	0.9422	1.0000e-04
28/50	22ms	0.1781	0.9352	0.1588	0.9422	1.0000e-04
29/50	22ms	0.1787	0.9429	0.1588	0.9422	1.0000e-05
30/50	21ms	0.1790	0.9319	0.1588	0.9422	1.0000e-05
31/50	21ms	0.1769	0.9343	0.1588	0.9422	1.0000e-05
32/50	21ms	0.1804	0.9338	0.1588	0.9422	1.0000e-05
33/50	21ms	0.1746	0.9371	0.1588	0.9422	1.0000e-05
34/50	21ms	0.1809	0.9390	0.1588	0.9422	1.0000e-05
35/50	21ms	0.1912	0.9290	0.1588	0.9422	1.0000e-05
36/50	21ms	0.1843	0.9362	0.1588	0.9422	1.0000e-05
37/50	22ms	0.1795	0.9390	0.1588	0.9422	2.0000e-06
38/50	22ms	0.1829	0.9338	0.1588	0.9422	2.0000e-06
39/50	21ms	0.1810	0.9367	0.1588	0.9422	2.0000e-06
40/50	24ms	0.1783	0.9343	0.1588	0.9422	2.0000e-06

4.3.2.4. Training Validation 20 Epoch (CLAHE – White Balance)

Berikut adalah hasil akurasi dan loss dari pelatihan dan validasi data dalam Gambar 4.22 untuk dua skenario, yaitu (a) dengan 3 kelas dan (b) dengan 2 kelas.



(a) (b)

Gambar 4.22 MobileNetV2 20 Epoch (CLAHE – White Balance). (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.22 dan tabel 4.41 dapat dilihat hasil pengujian pada 3 kelas, perhitungan loss dan akurasi pada data training dan data validasi. Pada epoch pertama, didapatkan loss sebesar 125,48% dan akurasi sebesar 46,44% pada data training, serta loss sebesar 69,47% dan akurasi sebesar 69,48% pada data validasi. Pada epoch selanjutnya, terjadi variasi dalam perubahan loss dan akurasi pada kedua dataset. Terdapat perubahan naik turun pada loss dan akurasi pada beberapa epoch. Pada akhir pelatihan, pada epoch ke 20, didapatkan loss sebesar 56,32% dan akurasi sebesar 74,35% pada data training, serta loss sebesar 55,06% dan akurasi sebesar 75,11% pada data validasi.

Tabel 4.41 Hasil MobileNetV2 20 Epoch 3 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/20	66ms	1.2548	0.4644	0.6947	0.6948	0.0010
2/20	33ms	0.8305	0.6314	0.6084	0.7467	0.0010
3/20	31ms	0.7555	0.6663	0.5948	0.7363	0.0010
4/20	31ms	0.7123	0.6727	0.6114	0.7156	0.0010
5/20	33ms	0.6698	0.6946	0.5668	0.7526	0.0010
6/20	33ms	0.6496	0.7098	0.5550	0.7541	0.0010
7/20	31ms	0.9213	0.7127	0.5610	0.7452	0.0010
8/20	31ms	0.6193	0.7162	0.5875	0.7244	0.0010
9/20	32ms	0.6154	0.7219	0.5635	0.7570	0.0010
10/20	32ms	0.5974	0.7349	0.7402	0.6489	0.0010
11/20	32ms	0.5888	0.7333	0.5494	0.7585	0.0010
12/20	32ms	0.5980	0.7279	0.5520	0.7600	0.0010
13/20	32ms	0.5870	0.7410	0.5805	0.7244	0.0010
14/20	31ms	0.5875	0.7390	0.5597	0.7467	0.0010
15/20	31ms	0.5784	0.7359	0.5566	0.7437	0.0010
16/20	31ms	0.5677	0.7508	0.5772	0.7289	0.0010
17/20	31ms	0.5645	0.7495	0.5767	0.7363	0.0010
18/20	31ms	0.5651	0.7568	0.5555	0.7570	0.0010

Tabel 4.41 Hasil MobileNetV2 20 Epoch 3 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
19/20	32ms	0.5827	0.7460	0.5381	0.7615	0.0010
20/20	31ms	0.5632	0.7435	0.5506	0.7511	0.0010

Sedangkan pada 2 kelas menunjukkan penurunan loss dan peningkatan akurasi pada setiap epoch. Pada epoch 20, model mencapai loss yang rendah dan akurasi yang tinggi pada data pelatihan dan validasi. Hasil ini menunjukkan bahwa model memiliki kemampuan prediksi yang baik seiring dengan peningkatan jumlah epoch. Pada epoch ke-20, didapatkan hasil loss sebesar 18,91% dan akurasi sebesar 92,86% pada data training, serta loss sebesar 15,29% dan akurasi sebesar 94,22% pada data validasi yang dapat dilihat pada gambar 4.22 dan tabel 4.42.

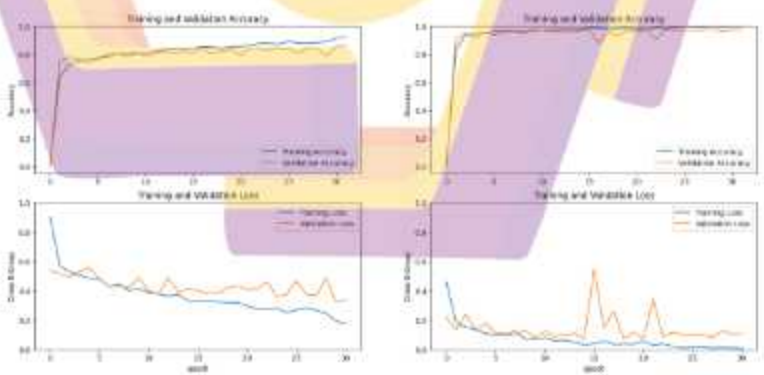
Tabel 4.42 Hasil MobileNetV2 20 Epoch 2 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/20	78ms	0.7924	0.6324	0.3100	0.8756	0.0010
2/20	34ms	0.4111	0.8186	0.2191	0.9178	0.0010
3/20	34ms	0.3225	0.8590	0.1984	0.9244	0.0010
4/20	35ms	0.3001	0.8729	0.1790	0.9356	0.0010
5/20	32ms	0.2553	0.8990	0.1788	0.9333	0.0010
6/20	32ms	0.2526	0.8981	0.1707	0.9333	0.0010
7/20	32ms	0.2383	0.9029	0.1822	0.9311	0.0010
8/20	33ms	0.2369	0.9081	0.1583	0.9444	0.0010
9/20	32ms	0.2360	0.9052	0.1678	0.9356	0.0010
10/20	32ms	0.2057	0.9195	0.1765	0.9267	0.0010
11/20	32ms	0.2110	0.9129	0.1534	0.9400	0.0010
12/20	32ms	0.1930	0.9210	0.1538	0.9400	0.0010
13/20	32ms	0.1948	0.9243	0.1802	0.9422	0.0010
14/20	32ms	0.1988	0.9233	0.1574	0.9400	0.0010
15/20	32ms	0.1730	0.9324	0.1666	0.9400	0.0010
16/20	33ms	0.1797	0.9290	0.1723	0.9400	0.0010
17/20	32ms	0.1799	0.9319	0.1528	0.9422	1.0000e-04

Tabel 4.42 Hasil MobileNetV2 20 Epoch 2 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
18/20	33ms	0.1915	0.9248	0.1532	0.9422	1.0000e-04
19/20	32ms	0.1903	0.9229	0.1526	0.9422	1.0000e-04
20/20	31ms	0.1891	0.9286	0.1529	0.9422	1.0000e-04

Peningkatan performa yang signifikan terlihat saat model dilatih dengan jumlah epoch yang lebih banyak. Terjadi penurunan drastis pada loss dan peningkatan yang signifikan pada akurasi pada setiap epoch. Pada akhir training, model mencapai akurasi tinggi dan loss rendah pada data training dan validasi yang menunjukkan prediksi baik. Setelah itu, dilakukan perbaikan hasil menggunakan fine tuning yang berfungsi untuk melakukan penyesuaian learning rate yang diatur dari mulai terkecil sampai terbesar dengan menambahkan 30 epoch tambahan untuk mengoptimalkan model. Perubahan hasil fine tuning ini dapat ditemukan di Gambar 4.23.



(a) (b)

Gambar 4.23 Fine Tuning MobileNetV2 20 Epoch (CLAHE – White Balance). (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.23 dan tavel 4.43 dapat terlihat bahwa hasil klasifikasi 3 kelas disimpulkan bahwa model mampu belajar dengan baik dan menunjukkan peningkatan kinerja seiring dengan peningkatan jumlah epoch. Akurasi training dan validasi meningkat dari epoch ke epoch, mencapai akurasi training 92,51% dan validasi sebesar 85,93% pada akhir training. Hal ini menunjukkan bahwa model tersebut mampu melakukan prediksi dengan baik pada data baru.

Tabel 4.43 Hasil Fine Tuning MobileNetV2 20 Epoch 3 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
20/50	121ms	0.9024	0.6365	0.5411	0.7526	1.0000e-04
21/50	76ms	0.5664	0.7286	0.5124	0.7763	1.0000e-04
22/50	75ms	0.5283	0.7530	0.4941	0.7704	1.0000e-04
23/50	76ms	0.5054	0.7638	0.5289	0.7378	1.0000e-04
24/50	75ms	0.4831	0.7749	0.5561	0.7674	1.0000e-04
25/50	77ms	0.4788	0.7797	0.4825	0.8015	1.0000e-04
26/50	77ms	0.4266	0.8063	0.4267	0.8030	1.0000e-04
27/50	76ms	0.4457	0.7892	0.4341	0.8044	1.0000e-04
28/50	77ms	0.3995	0.8054	0.4179	0.8148	1.0000e-04
29/50	75ms	0.4139	0.8025	0.4905	0.7867	1.0000e-04
30/50	76ms	0.3886	0.8251	0.3973	0.8030	1.0000e-04
31/50	77ms	0.3764	0.8308	0. 3774	0.8222	1.0000e-04
32/50	75ms	0.3635	0.8381	0. 4888	0.8044	1.0000e-04
33/50	76ms	0.3718	0.8346	0.3887	0.8222	1.0000e-04
34/50	75ms	0.3316	0.8448	0.4110	0.8133	1.0000e-04
35/50	77ms	0.3238	0.8543	0.4005	0.8444	1.0000e-04
36/50	75ms	0.3322	0.8514	0.3811	0.8015	1.0000e-04
37/50	75ms	0.3231	0.8432	0.3830	0.8267	1.0000e-04
38/50	76ms	0.3200	0.8543	0.4187	0.8311	1.0000e-04
39/50	76ms	0.3189	0.8568	0.4334	0.7926	1.0000e-04
40/50	77ms	0.2950	0.8622	0.4077	0.8474	1.0000e-04
41/50	75ms	0.2779	0.8794	0.4126	0.8341	1.0000e-04
42/50	76ms	0.2720	0.8784	0.4623	0.8311	1.0000e-04
43/50	76ms	0.2808	0.8737	0.3585	0.8415	1.0000e-04
44/50	75ms	0.2495	0.8921	0.3741	0.8415	1.0000e-04

Tabel 4.43 Hasil Fine Tuning MobileNetV2 20 Epoch 3 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
45/50	76ms	0.2694	0.8810	0.4687	0.8104	1.0000e-04
46/50	75ms	0.2796	0.8781	0.3748	0.8430	1.0000e-04
47/50	76ms	0.2667	0.8841	0.3655	0.8444	1.0000e-04
48/50	76ms	0.2452	0.8902	0.4886	0.7911	1.0000e-04
49/50	77ms	0.1934	0.9137	0.3225	0.8578	1.0000e-05
50/50	77ms	0.1726	0.9251	0.3395	0.8593	1.0000e-05

Sedangkan pengujian pada 2 kelas dapat dilihat juga pada gambar tersebut, bahwa model mengalami peningkatan akurasi seiring dengan peningkatan jumlah epoch. Pada awal training, akurasi validasi meningkat secara signifikan dari 91,11% menjadi 95,11%. Selanjutnya, akurasi validasi tetap tinggi dengan fluktuasi kecil, menunjukkan bahwa model telah mencapai tingkat kestabilan dalam melakukan prediksi. Selain itu, model juga memiliki akurasi training yang tinggi, mencapai 99,62% pada epoch 20 yang dapat dilihat pada gambar 4.23 dam tabel 4.44. Hal ini menunjukkan bahwa model mampu belajar dengan baik dari data pelatihan.

Tabel 4.44 Hasil Fine Tuning MobileNetV2 20 Epoch 2 Kelas (CLAHE – White Balance)

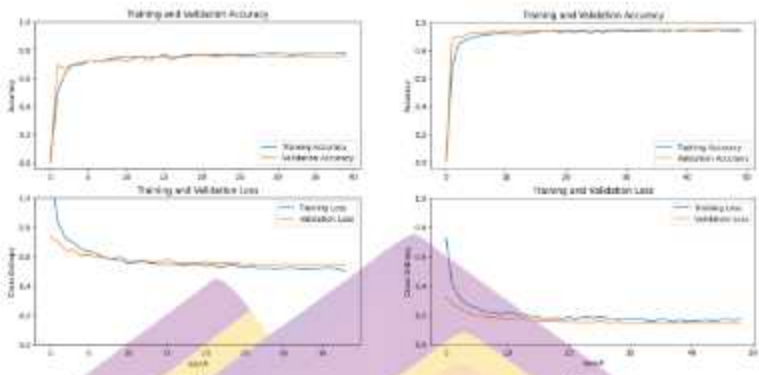
Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
20/50	149ms	0.4625	0.8338	0.2174	0.9111	1.0000e-04
21/50	77ms	0.1806	0.9376	0.1371	0.9511	1.0000e-04
22/50	76ms	0.1529	0.9443	0.2408	0.9178	1.0000e-04
23/50	78ms	0.1343	0.9514	0.1323	0.9578	1.0000e-04
24/50	76ms	0.1095	0.9633	0.1829	0.9356	1.0000e-04
25/50	76ms	0.1010	0.9638	0.1115	0.9578	1.0000e-04
26/50	78ms	0.0937	0.9671	0.1084	0.9644	1.0000e-04
27/50	76ms	0.1262	0.9524	0.0969	0.9644	1.0000e-04
28/50	76ms	0.0707	0.9738	0.1339	0.9644	1.0000e-04
29/50	78ms	0.0691	0.9738	0.0790	0.9733	1.0000e-04
30/50	76ms	0.0820	0.9729	0.1234	0.9600	1.0000e-04

Tabel 4.44 Hasil Fine Tuning MobileNetV2 20 Epoch 2 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
31/50	75ms	0.0598	0.9790	0.0875	0.9689	1.0000e-04
32/50	75ms	0.0634	0.9786	0.1023	0.9600	1.0000e-04
33/50	76ms	0.0494	0.9795	0.1099	0.9667	1.0000e-04
34/50	78ms	0.0329	0.9890	0.0746	0.9756	1.0000e-04
35/50	76ms	0.0389	0.9881	0.5451	0.8778	1.0000e-04
36/50	75ms	0.0569	0.9805	0.1499	0.9622	1.0000e-04
37/50	75ms	0.0349	0.9857	0.2599	0.9333	1.0000e-04
38/50	76ms	0.0374	0.9886	0.0938	0.9733	1.0000e-04
39/50	76ms	0.0363	0.9843	0.1130	0.9600	1.0000e-04
40/50	76ms	0.0573	0.9771	0.0767	0.9733	1.0000e-04
41/50	75ms	0.0295	0.9914	0.3426	0.9067	1.0000e-04
42/50	76ms	0.0393	0.9838	0.0867	0.9756	1.0000e-04
43/50	75ms	0.0217	0.9914	0.1163	0.9689	1.0000e-05
44/50	77ms	0.0096	0.9971	0.1012	0.9711	1.0000e-05
45/50	76ms	0.0138	0.9918	0.1047	0.9711	1.0000e-05
46/50	76ms	0.0149	0.9943	0.1015	0.9711	1.0000e-05
47/50	78ms	0.0098	0.9967	0.0802	0.9778	1.0000e-05
48/50	75ms	0.0114	0.9952	0.1285	0.9622	1.0000e-05
49/50	76ms	0.0102	0.9967	0.1037	0.9733	1.0000e-05
50/50	75ms	0.0086	0.9962	0.1083	0.9711	1.0000e-05

4.3.2.5. Training Validation 50 Epoch

Berikut adalah hasil akurasi dan loss dari pelatihan dan validasi data dalam Gambar 4.24 untuk dua skenario, yaitu (a) dengan 3 kelas dan (b) dengan 2 kelas.



(a) (b)

Gambar 4.24 MobileNetV2 50 Epoch. (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Hasil pada klasifikasi 3 kelas dapat dilihat pada gambar 4.24 dan tabel 4.45, hasil tersebut mampu meningkatkan akurasi dan mengurangi loss pada setiap epoch. Awalnya, akurasi model sekitar 50,48%, namun dengan berjalannya epoch, akurasi meningkat menjadi sekitar 78,13% pada data training. Pada data validasi, akurasi juga mengalami peningkatan, dari sekitar 68,89% menjadi sekitar 75,85%. Hal ini menunjukkan bahwa model memiliki kemampuan untuk mengklasifikasikan data dengan akurasi yang wajar. Meskipun terjadi fluktuasi dalam akurasi dan loss pada beberapa epoch, perubahan tersebut tidak signifikan. Penurunan learning rate yang dilakukan juga berdampak positif dalam meningkatkan akurasi model.

Tabel 4.45 Hasil MobileNetV2 50 Epoch 3 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/50	46ms	1.1654	0.5048	0.7319	0.6889	0.0010
2/50	21ms	0.8288	0.6390	0.7009	0.6681	0.0010
3/50	22ms	0.7213	0.6917	0.6419	0.6948	0.0010
4/50	22ms	0.6939	0.6943	0.6416	0.7081	0.0010

Tabel 4.45 Hasil MobileNetV2 50 Epoch 3 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
5/50	23ms	0.6466	0.7175	0.6052	0.7185	0.0010
6/50	21ms	0.6344	0.7171	0.6100	0.7126	0.0010
7/50	23ms	0.6162	0.7194	0.5918	0.7259	0.0010
8/50	21ms	0.5929	0.7365	0.5897	0.7200	0.0010
9/50	23ms	0.5823	0.7457	0.5841	0.7393	0.0010
10/50	21ms	0.5781	0.7429	0.5979	0.7141	0.0010
11/50	21ms	0.5536	0.7517	0.5735	0.7378	0.0010
12/50	22ms	0.5607	0.7514	0.5726	0.7481	0.0010
13/50	20ms	0.5653	0.7511	0.5726	0.7230	0.0010
14/50	22ms	0.5553	0.7514	0.5607	0.7496	0.0010
15/50	21ms	0.5476	0.7603	0.5607	0.7511	0.0010
16/50	21ms	0.5509	0.7543	0.5826	0.7274	0.0010
17/50	21ms	0.5474	0.7524	0.5576	0.7481	0.0010
18/50	21ms	0.5365	0.7603	0.5535	0.7452	0.0010
19/50	22ms	0.5425	0.7641	0.5536	0.7659	0.0010
20/50	21ms	0.5455	0.7584	0.5551	0.7600	0.0010
21/50	22ms	0.5370	0.7651	0.5626	0.7615	0.0010
22/50	22ms	0.5372	0.7610	0.5547	0.7422	0.0010
23/50	22ms	0.5285	0.7613	0.5557	0.7630	0.0010
24/50	22ms	0.5411	0.7616	0.5552	0.7630	0.0010
25/50	21ms	0.5369	0.7641	0.5536	0.7644	0.0010
26/50	21ms	0.5236	0.7625	0.5484	0.7541	0.0010
27/50	21ms	0.5379	0.7660	0.5429	0.7467	0.0010
28/50	21ms	0.5168	0.7743	0.5444	0.7556	1.0000e-04
29/50	21ms	0.5235	0.7771	0.5431	0.7511	1.0000e-04
30/50	21ms	0.5123	0.7759	0.5429	0.7570	1.0000e-04
31/50	20ms	0.5212	0.7695	0.5471	0.7422	1.0000e-04
32/50	21ms	0.5250	0.7632	0.5425	0.7556	1.0000e-04
33/50	21ms	0.5148	0.7733	0.5427	0.7570	1.0000e-04
34/50	21ms	0.5142	0.7705	0.5428	0.7585	1.0000e-04
35/50	21ms	0.5173	0.7695	0.5464	0.7437	1.0000e-04
36/50	20ms	0.5204	0.7724	0.5448	0.7467	1.0000e-05
37/50	20ms	0.5266	0.7711	0.5436	0.7526	1.0000e-05
38/50	21ms	0.5116	0.7768	0.5433	0.7541	1.0000e-05
39/50	23ms	0.5041	0.7813	0.5431	0.7585	1.0000e-05

Sedangkan pada gambar tersebut juga dapat dilihat hasil klasifikasi 2 kelas, berdasarkan grafik yang ditampilkan dapat dilihat bahwa selama training, akurasi model meningkat sedangkan loss menurun. Ini menunjukkan bahwa model secara bertahap mempelajari pola yang ada dalam data training. Pada epoch pertama, akurasi model adalah 66,62%, tetapi meningkat secara signifikan menjadi 94,14% pada epoch ke 48 dan loss model turun dari 72,80% menjadi 16,62% pada epoch terakhir, sedangkan pada dataset validation model berhasil mencapai akurasi sekitar 94,89% yang dapat dilihat pada tabel 4.46.

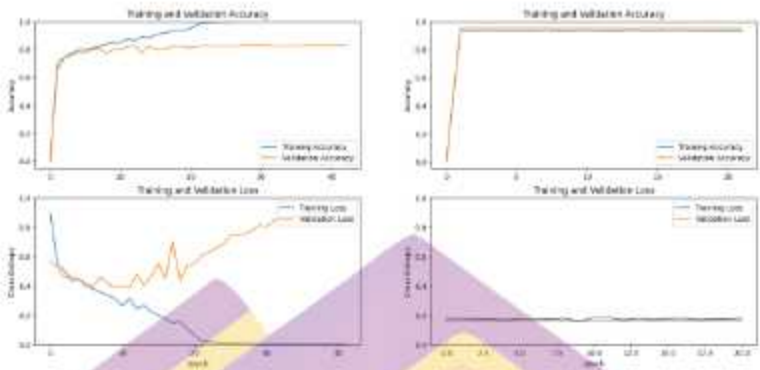
Tabel 4.46 Hasil MobileNetV2 50 Epoch 2 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/50	59ms	0.7280	0.6662	0.3182	0.8844	0.0010
2/50	24ms	0.4033	0.8381	0.2750	0.9000	0.0010
3/50	24ms	0.3327	0.8705	0.2498	0.9022	0.0010
4/50	24ms	0.2844	0.8862	0.2268	0.9244	0.0010
5/50	24ms	0.2609	0.8995	0.2072	0.9267	0.0010
6/50	22ms	0.2498	0.9062	0.1983	0.9222	0.0010
7/50	24ms	0.2274	0.9119	0.1914	0.9333	0.0010
8/50	22ms	0.2174	0.9171	0.1830	0.9333	0.0010
9/50	22ms	0.2129	0.9238	0.1795	0.9289	0.0010
10/50	24ms	0.2035	0.9233	0.1795	0.9378	0.0010
11/50	22ms	0.2167	0.9171	0.1725	0.9333	0.0010
12/50	22ms	0.2127	0.9148	0.1674	0.9356	0.0010
13/50	22ms	0.1910	0.9195	0.1675	0.9333	0.0010
14/50	22ms	0.1935	0.9314	0.1663	0.9356	0.0010
15/50	21ms	0.1947	0.9295	0.1662	0.9378	0.0010
16/50	22ms	0.1816	0.9381	0.1600	0.9378	0.0010
17/50	22ms	0.1765	0.9386	0.1579	0.9378	0.0010
18/50	24ms	0.1771	0.9381	0.1567	0.9422	0.0010
19/50	22ms	0.1812	0.9295	0.1565	0.9400	0.0010
20/50	22ms	0.1724	0.9357	0.1548	0.9400	0.0010
21/50	22ms	0.1858	0.9319	0.1527	0.9400	0.0010
22/50	22ms	0.1750	0.9338	0.1541	0.9422	0.0010

Tabel 4.46 Hasil MobileNetV2 50 Epoch 2 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
23/50	25ms	0.1812	0.9371	0.1498	0.9444	0.0010
24/50	24ms	0.1846	0.9276	0.1523	0.9489	0.0010
25/50	22ms	0.1785	0.9376	0.1499	0.9400	0.0010
26/50	21ms	0.1849	0.9300	0.1601	0.9467	0.0010
27/50	22ms	0.1818	0.9362	0.1518	0.9444	0.0010
28/50	22ms	0.1691	0.9400	0.1476	0.9444	0.0010
29/50	24ms	0.1671	0.9400	0.1549	0.9511	0.0010
30/50	22ms	0.1766	0.9348	0.1481	0.9444	0.0010
31/50	22ms	0.17055	0.9400	0.1454	0.9444	0.0010
32/50	22ms	0.1691	0.9367	0.1440	0.9489	0.0010
33/50	21ms	0.1719	0.9362	0.1469	0.9489	0.0010
34/50	22ms	0.1569	0.9400	0.1482	0.9489	0.0010
35/50	21ms	0.1577	0.9448	0.1453	0.9467	0.0010
36/50	21ms	0.1727	0.9376	0.1431	0.9467	0.0010
37/50	22ms	0.1641	0.9390	0.1468	0.9489	0.0010
38/50	22ms	0.1519	0.9471	0.1418	0.9467	1.0000e-04
39/50	22ms	0.1674	0.9367	0.1419	0.9467	1.0000e-04
40/50	22ms	0.1541	0.9438	0.1417	0.9489	1.0000e-04
41/50	22ms	0.1594	0.9443	0.1416	0.9467	1.0000e-04
42/50	22ms	0.1586	0.9405	0.1418	0.9489	1.0000e-04
43/50	22ms	0.1647	0.9376	0.1418	0.9467	1.0000e-04
44/50	21ms	0.1614	0.9419	0.1417	0.9489	1.0000e-04
45/50	22ms	0.1590	0.9462	0.1412	0.9489	1.0000e-04
46/50	22ms	0.1672	0.9410	0.1412	0.9489	1.0000e-05
47/50	22ms	0.1662	0.9414	0.1411	0.9489	1.0000e-05
48/50	23ms	0.1625	0.9376	0.1411	0.9489	1.0000e-05
49/50	26ms	0.1680	0.9376	0.1411	0.9489	1.0000e-05

Setelah itu, peneliti mengoptimalkan model dengan menggunakan fine tuning yang berfungsi untuk melakukan penyesuaian learning rate yang diatur dari mulai terkecil sampai terbesar dengan menambahkan 30 epoch tambahan. Perubahan hasil finetuning ini dapat ditemukan di Gambar 4.25.



(a) (b)

Gambar 4.25 Fine Tuning MobileNetV2 50 Epoch. (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.25 dan tabel 4.47 dapat dilihat hasil dari 3 kelas setelah tahap finetuning, sehingga disimpulkan bahwa pelatihan model memiliki hasil yang baik. Pada awalnya, model memiliki akurasi sekitar 66% pada data training, namun dengan berjalannya waktu, akurasi meningkat secara signifikan hingga mencapai sekitar 99,81%. Model juga mampu mengurangi loss secara konsisten seiring dengan peningkatan akurasi. Pada data validasi, model awalnya memiliki akurasi sekitar 71,11% dan loss sekitar 56,77%. Namun dengan meningkatnya jumlah 50 epoch, model berhasil meningkatkan akurasi pada data validasi menjadi sekitar 83,11% dengan loss sekitar 91,02%.

Tabel 4.47 Hasil Fine Tuning MobileNetV2 50 Epoch 3 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
39/80	46ms	0.8936	0.6600	0.5677	0.7111	1.0000e-04
40/80	21ms	0.5529	0.7416	0.5252	0.7393	1.0000e-04
41/80	22ms	0.5001	0.7717	0.4538	0.7556	1.0000e-04
42/80	22ms	0.4321	0.7956	0.4689	0.7807	1.0000e-04
43/80	23ms	0.4479	0.7949	0.4379	0.7778	1.0000e-04
44/80	21ms	0.3947	0.8092	0.4160	0.7985	1.0000e-04

Tabel 4.47 Hasil Fine Tuning MobileNetV2 50 Epoch 3 Kelas (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
45/80	23ms	0.3789	0.8200	0.3979	0.8074	1.0000e-04
46/80	21ms	0.3575	0.8384	0.4610	0.7704	1.0000e-04
47/80	23ms	0.3375	0.8467	0.4115	0.8000	1.0000e-04
48/80	21ms	0.3140	0.8505	0.3966	0.8015	1.0000e-04
49/80	21ms	0.2635	0.8737	0.3994	0.8178	1.0000e-04
50/80	22ms	0.3131	0.8638	0.3893	0.8267	1.0000e-04
51/80	20ms	0.2440	0.8914	0.4773	0.7763	1.0000e-04
52/80	22ms	0.2678	0.8813	0.4062	0.8207	1.0000e-04
53/80	21ms	0.2240	0.9051	0.5691	0.8015	1.0000e-04
54/80	21ms	0.2002	0.9156	0.5538	0.8015	1.0000e-04
55/80	21ms	0.1771	0.9235	0.4467	0.8148	1.0000e-04
56/80	21ms	0.1474	0.9403	0.7047	0.8267	1.0000e-04
57/80	22ms	0.1590	0.9330	0.4334	0.8163	1.0000e-04
58/80	21ms	0.1093	0.9521	0.5364	0.8104	1.0000e-04
59/80	22ms	0.0635	0.9775	0.5528	0.8222	1.0000e-05
60/80	23ms	0.0268	0.9921	0.6071	0.8281	1.0000e-05
61/80	21ms	0.0201	0.9937	0.6259	0.8267	1.0000e-05
62/80	23ms	0.0154	0.9978	0.6569	0.8237	1.0000e-05
63/80	21ms	0.0143	0.9968	0.6823	0.8252	1.0000e-05
64/80	23ms	0.0104	0.9981	0.7413	0.8252	1.0000e-05
65/80	21ms	0.0074	0.9994	0.7453	0.8267	1.0000e-05
66/80	21ms	0.0080	0.9981	0.7525	0.8326	1.0000e-05
67/80	22ms	0.0055	1.0000	0.7761	0.8296	1.0000e-05
68/80	20ms	0.0053	0.9990	0.8119	0.8311	1.0000e-05
69/80	22ms	0.0041	0.9997	0.8034	0.8281	1.0000e-05
70/80	21ms	0.0034	0.9997	0.8434	0.8267	1.0000e-05
71/80	21ms	0.0028	1.0000	0.8665	0.8267	1.0000e-05
72/80	21ms	0.0024	1.0000	0.8783	0.8252	1.0000e-05
73/80	22ms	0.0028	0.9997	0.8632	0.8296	1.0000e-05
74/80	20ms	0.0018	1.0000	0.8884	0.8252	1.0000e-05
75/80	22ms	0.0017	1.0000	0.8943	0.8311	2.0000e-06
76/80	21ms	0.0021	0.9997	0.8957	0.8281	2.0000e-06
77/80	21ms	0.0018	1.0000	0.8978	0.8296	2.0000e-06
78/80	22ms	0.0015	1.0000	0.9115	0.8267	2.0000e-06
79/80	21ms	0.0018	1.0000	0.9103	0.8311	2.0000e-06
80/80	22ms	0.0015	1.0000	0.9102	0.8311	2.0000e-06

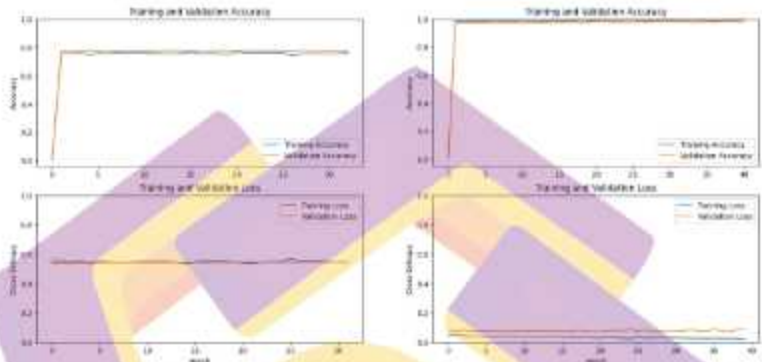
Sedangkan pada klasifikasi 2 kelas juga dapat dilihat pada gambar tersebut dengan hasil yang ditemukan bahwa nilai loss berkurang secara bertahap, sedangkan akurasi meningkat. Pada tahap validasi, model juga mencapai hasil yang baik dengan loss dan akurasi yang stabil. Pada epoch 50, model mencapai loss sebesar 15,49% dan akurasi sebesar 95,11% pada data validasi yang dapat dilihat pada tabel 4.48.

Tabel 4.48 Hasil Fine Tuning MobileNetV2 50 Epoch 2 Kelas

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
49/80	54ms	0.1770	0.9348	0.1549	0.9511	1.0000e-04
50/80	21ms	0.1776	0.9333	0.1549	0.9511	1.0000e-04
51/80	21ms	0.1752	0.9352	0.1549	0.9511	1.0000e-04
52/80	21ms	0.1745	0.9333	0.1549	0.9511	1.0000e-04
53/80	21ms	0.1687	0.9348	0.1549	0.9511	1.0000e-04
54/80	21ms	0.1762	0.9310	0.1549	0.9511	1.0000e-04
55/80	21ms	0.1773	0.9376	0.1549	0.9511	1.0000e-04
56/80	21ms	0.1733	0.9300	0.1549	0.9511	1.0000e-04
57/80	21ms	0.1782	0.9314	0.1549	0.9511	1.0000e-04
58/80	21ms	0.1618	0.9371	0.1549	0.9511	1.0000e-05
59/80	21ms	0.1826	0.9324	0.1549	0.9511	1.0000e-05
60/80	21ms	0.1815	0.9267	0.1549	0.9511	1.0000e-05
61/80	21ms	0.1692	0.9386	0.1549	0.9511	1.0000e-05
62/80	21ms	0.1790	0.9333	0.1549	0.9511	1.0000e-05
63/80	21ms	0.1709	0.9338	0.1549	0.9511	1.0000e-05
64/80	22ms	0.1760	0.9352	0.1549	0.9511	1.0000e-05
65/80	22ms	0.1780	0.9338	0.1549	0.9511	1.0000e-05
66/80	21ms	0.1709	0.9324	0.1549	0.9511	2.0000e-06
67/80	21ms	0.1742	0.9314	0.1549	0.9511	2.0000e-06
68/80	21ms	0.1742	0.9286	0.1549	0.9511	2.0000e-06
69/80	24ms	0.1788	0.9300	0.1549	0.9511	2.0000e-06

4.3.2.6. Training Validation 50 Epoch (CLAHE – White Balance)

Berikut adalah hasil akurasi dan loss dari pelatihan dan validasi data dalam Gambar 4.26 untuk dua skenario, yaitu (a) dengan 3 kelas dan (b) dengan 2 kelas.



(a) (b)

Gambar 4.26 MobileNetV2 50 Epoch (CLAHE – White Balance). (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Pada gambar 4.26 dan tabel 4.49 dapat dilihat bahwa klasifikasi dengan 3 kelas, model mengalami peningkatan akurasi dan penurunan tingkat loss selama proses training. Pada epoch 50, model mencapai akurasi sekitar 73% pada data training dan sekitar 83% pada data validasi. Selama pelatihan, dilakukan penyesuaian tingkat pembelajaran (learning rate) menggunakan metode "ReduceLROnPlateau" untuk meningkatkan performa model.

Tabel 4.49 Hasil MobileNetV2 50 Epoch 3 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/50	48ms	0.5601	0.7562	0.5386	0.7733	2.0000e-06
2/50	33ms	0.5586	0.7552	0.8390	0.7719	2.0000e-06
3/50	33ms	0.5428	0.7600	0.5393	0.7733	2.0000e-06
4/50	33ms	0.5552	0.7460	0.5400	0.7733	2.0000e-06
5/50	28ms	0.5345	0.7590	0.5403	0.7733	2.0000e-06
6/50	27ms	0.5544	0.7552	0.5407	0.7733	2.0000e-06

Tabel 4.49 Hasil MobileNetV2 50 Epoch 3 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
7/50	27ms	0.5438	0.7625	0.5411	0.7719	2.0000e-06
8/50	25ms	0.5416	0.7578	0.5413	0.7719	2.0000e-06
9/50	25ms	0.5430	0.7635	0.5416	0.7733	2.0000e-06
10/50	23ms	0.5485	0.7625	0.5418	0.7748	2.0000e-06
11/50	23ms	0.5483	0.7687	0.5418	0.7748	2.0000e-06
12/50	27ms	0.5495	0.7559	0.5419	0.7763	2.0000e-06
13/50	25ms	0.5504	0.7527	0.5420	0.7748	2.0000e-06
14/50	25ms	0.5431	0.7578	0.5422	0.7748	2.0000e-06
15/50	23ms	0.5351	0.7660	0.5422	0.7748	2.0000e-06
16/50	24ms	0.5454	0.7556	0.5424	0.7748	2.0000e-06
17/50	22ms	0.5550	0.7571	0.5425	0.7748	2.0000e-06
18/50	23ms	0.5522	0.7571	0.5428	0.7748	2.0000e-06
19/50	23ms	0.5500	0.7473	0.5431	0.7748	2.0000e-06
20/50	23ms	0.5458	0.7683	0.5431	0.7748	2.0000e-06
21/50	21ms	0.5391	0.7603	0.5431	0.7748	2.0000e-06
22/50	23ms	0.5364	0.7603	0.5433	0.7748	2.0000e-06
23/50	23ms	0.5433	0.7622	0.5433	0.7748	2.0000e-06
24/50	23ms	0.5473	0.7632	0.5433	0.7768	2.0000e-06
25/50	25ms	0.5483	0.7606	0.5436	0.7733	2.0000e-06
26/50	28ms	0.5686	0.7397	0.5439	0.7733	2.0000e-06
27/50	26ms	0.5511	0.7543	0.5440	0.7748	2.0000e-06
28/50	24ms	0.5548	0.7565	0.5440	0.7748	2.0000e-06
29/50	23ms	0.5519	0.7546	0.5442	0.7748	2.0000e-06
30/50	23ms	0.5482	0.7549	0.5443	0.7733	2.0000e-06
31/50	22ms	0.5447	0.7543	0.5445	0.7733	2.0000e-06
32/50	23ms	0.5418	0.7629	0.5445	0.7719	2.0000e-06

Sedangkan untuk hasil pada klasifikasi 2 kelas dapat dilihat juga pada gambar tersebut, bahwa model tersebut memiliki tingkat loss yang rendah dan akurasi yang tinggi. Pada epoch 50, model mencapai loss sekitar 2,25% dan akurasi sekitar 99,24% pada data training. Sedangkan pada data validasi, model mencapai

loss sekitar 8,59% dan akurasi sekitar 98,00% yang dapat dilihat pada gambar 4.26 dan tabel 4.50.

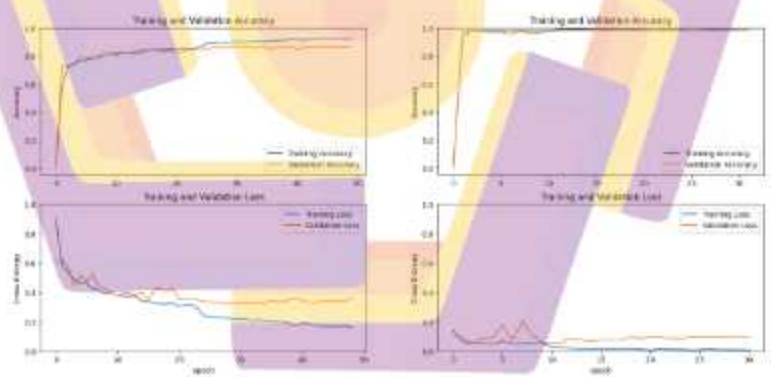
Tabel 4.50 Hasil MobileNetV2 50 Epoch 2 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
1/50	47ms	0.0463	0.9871	0.0819	0.9733	2.0000e-06
2/50	22ms	0.0501	0.9824	0.0754	0.9733	2.0000e-06
3/50	21ms	0.0439	0.9838	0.0818	0.9733	2.0000e-06
4/50	22ms	0.0419	0.9843	0.0761	0.9733	2.0000e-06
5/50	22ms	0.0431	0.9833	0.0766	0.9733	2.0000e-06
6/50	22ms	0.0397	0.9890	0.0761	0.9733	2.0000e-06
7/50	22ms	0.0384	0.9852	0.0747	0.9733	2.0000e-06
8/50	21ms	0.0429	0.9862	0.0767	0.9733	2.0000e-06
9/50	21ms	0.0427	0. 9843	0.0751	0.9756	2.0000e-06
10/50	21ms	0.0368	0.9895	0.0774	0.9733	2.0000e-06
11/50	22ms	0.0393	0.9862	0.0781	0.9733	2.0000e-06
12/50	21ms	0.0394	0.9871	0.0768	0.9733	2.0000e-06
13/50	22ms	0.0378	0.9838	0.0780	0.9733	2.0000e-06
14/50	22ms	0.0339	0.9876	0.0759	0.9778	2.0000e-06
15/50	22ms	0.0417	0.9833	0.0783	0.9778	2.0000e-06
16/50	22ms	0.0377	0.9862	0.0744	0.9778	2.0000e-06
17/50	21ms	0.0340	0.9876	0.0745	0.9800	2.0000e-06
18/50	21ms	0.0320	0.9905	0.0814	0.9756	2.0000e-06
19/50	21ms	0.0338	0.9876	0.0773	0.9778	2.0000e-06
20/50	21ms	0.0306	0.9886	0.0734	0.9822	2.0000e-06
21/50	21ms	0.0290	0.9910	0.0726	0.9822	2.0000e-06
22/50	21ms	0.0311	0.9905	0.0779	0.9800	2.0000e-06
23/50	21ms	0.0313	0.9881	0.0806	0.9778	2.0000e-06
24/50	21ms	0.0271	0.9905	0.0793	0.9778	2.0000e-06
25/50	21ms	0.0260	0.9919	0.0847	0.9778	2.0000e-06
26/50	21ms	0.0329	0.9871	0.0766	0.9822	2.0000e-06
27/50	22ms	0.0281	0.9886	0.0829	0.9778	2.0000e-06
28/50	21ms	0.0296	0.9881	0.0732	0.9822	2.0000e-06
29/50	22ms	0.0308	0.9871	0.0761	0.9822	2.0000e-06
30/50	22ms	0.0277	0.9905	0.0730	0.9822	2.0000e-06
31/50	22ms	0.0301	0.9881	0.0744	0.9822	2.0000e-06
32/50	22ms	0.0288	0.9900	0.0784	0.9800	2.0000e-06

Tabel 4.50 Hasil MobileNetV2 50 Epoch 2 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
33/50	21ms	0.0266	0.9905	0.0837	0.9778	2.0000e-06
34/50	21ms	0.0277	0.9895	0.0778	0.9800	2.0000e-06
35/50	21ms	0.0260	0.9910	0.0742	0.9822	2.0000e-06
36/50	21ms	0.0273	0.9857	0.0860	0.9778	2.0000e-06
37/50	21ms	0.0267	0.9890	0.0786	0.9822	2.0000e-06
38/50	21ms	0.0258	0.9905	0.0723	0.9822	2.0000e-06
39/50	20ms	0.0257	0.9910	0.0833	0.9800	2.0000e-06
40/50	21ms	0.0225	0.9924	0.0859	0.9800	2.0000e-06

Setelah itu, peneliti mengoptimalkan model dengan menggunakan fine tuning yang berfungsi untuk melakukan penyesuaian learning rate yang diatur dari mulai terkecil sampai terbesar dengan menambahkan 30 epoch tambahan. Perubahan hasil finetuning ini dapat ditemukan di Gambar 4.27.



(a) (b)

Gambar 4.27 Fine Tuning MobileNetV2 50 Epoch (CLAHE – White Balance). (a) Training Validation 3 kelas dan (b) Training Validation 2 kelas

Training dan validation finetuning pada klasifikasi 3 kelas dapat dilihat pada gambar 4.27 dan tabel 4.51, hasil akurasi training sebesar 92.67% dan validasi sebesar 86.52% pada 50 epoch. Selain itu, juga learning rate dikurangi secara

otomatis menggunakan ReduceLROnPlateau callback saat validation loss tidak mengalami perbaikan. Kesimpulannya adalah model tersebut berhasil mencapai akurasi yang cukup tinggi pada data pelatihan dan validasi.

Tabel 4.51 Hasil Fine Tuning MobileNetV2 50 Epoch 3 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
32/80	48ms	0.9084	0.5752	0.6105	0.7244	1.0000e-04
33/80	36ms	0.5854	0.7213	0.6272	0.7363	1.0000e-04
34/80	28ms	0.8357	0.7460	0.5694	0.7319	1.0000e-04
35/80	24ms	0.4879	0.7638	0.4445	0.8000	1.0000e-04
36/80	24ms	0.4670	0.7717	0.5181	0.7778	1.0000e-04
37/80	23ms	0.4618	0.7829	0.4717	0.7896	1.0000e-04
38/80	23ms	0.4287	0.7946	0.5276	0.7837	1.0000e-04
39/80	27ms	0.4164	0.8108	0.4380	0.8133	1.0000e-04
40/80	25ms	0.389	0.8181	0.4284	0.8207	1.0000e-04
41/80	25ms	0.4008	0.8083	0.4014	0.8252	1.0000e-04
42/80	23ms	0.3774	0.8270	0.3741	0.8296	1.0000e-04
43/80	24ms	0.3749	0.8190	0.3975	0.8193	1.0000e-04
44/80	22ms	0.3741	0.8305	0.3933	0.8133	1.0000e-04
45/80	23ms	0. 3757	0.8257	0.3837	0.8415	1.0000e-04
46/80	23ms	0.3410	0.8441	0.4007	0.8341	1.0000e-04
47/80	23ms	0.3410	0.8435	0.3400	0.8519	1.0000e-04
48/80	23ms	0.3266	0.8463	0.4248	0.8326	1.0000e-04
49/80	23ms	0.3213	0.8486	0.4229	0.8326	1.0000e-04
50/80	27ms	0.3210	0.8556	0.4082	0.8370	1.0000e-04
51/80	25ms	0.3257	0.8514	0.4350	0.8193	1.0000e-04
52/80	25ms	0.3046	0.8556	0.3515	0.8326	1.0000e-04
53/80	23ms	0.3141	0.8530	0.3409	0.8474	1.0000e-04
54/80	24ms	0.3173	0.8530	0.3569	0.8444	1.0000e-04
55/80	22ms	0.2928	0.8695	0.3498	0.8489	1.0000e-04
56/80	23ms	0.2391	0.8902	0.3392	0.8548	1.0000e-05
57/80	23ms	0.2389	0.8889	0.3345	0.8607	1.0000e-05
58/80	23ms	0.2274	0.8990	0.3312	0.8593	1.0000e-05
59/80	27ms	0.2260	0.8959	0.3321	0.8637	1.0000e-05

Tabel 4.51 Hasil Fine Tuning MobileNetV2 50 Epoch 3 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
60/80	25ms	0.2263	0.8981	0.3207	0.8637	1.0000e-05
61/80	25ms	0.2169	0.9048	0.3272	0.8681	1.0000e-05
62/80	23ms	0.2215	0.9016	0.3330	0.8667	1.0000e-05
63/80	24ms	0.2175	0.9016	0.3221	0.8667	1.0000e-05
64/80	22ms	0.2082	0.9076	0.3315	0.8548	1.0000e-05
65/80	23ms	0.2166	0.9041	0.3298	0.8711	1.0000e-05
66/80	23ms	0.2040	0.9063	0.3315	0.8593	1.0000e-05
67/80	23ms	0.2067	0.9073	0.3470	0.8548	1.0000e-05
68/80	23ms	0.1963	0.9117	0.3365	0.8652	1.0000e-05
69/80	23ms	0.2016	0.9140	0.3399	0.8593	1.0000e-05
70/80	24ms	0.1852	0.9165	0.3523	0.8637	1.0000e-05
71/80	22ms	0.1798	0.9203	0.3623	0.8489	1.0000e-05
72/80	23ms	0.1890	0.9114	0.3437	0.8622	1.0000e-05
73/80	23ms	0.1889	0.9187	0.3333	0.8607	1.0000e-05
74/80	23ms	0.1763	0.9200	0.3307	0.8681	2.0000e-06
75/80	23ms	0.1692	0.9308	0.3405	0.8681	2.0000e-06
76/80	24ms	0.1716	0.9267	0.3401	0.8622	2.0000e-06
77/80	22ms	0.1690	0.9229	0.3459	0.8652	2.0000e-06
78/80	23ms	0.1694	0.9244	0.3410	0.8652	2.0000e-06
79/80	22ms	0.1708	0.9216	0.3460	0.8652	2.0000e-06
80/80	23ms	0.1657	0.9267	0.3516	0.8652	2.0000e-06

Sedangkan fine tuning pada 2 kelas juga dapat dilihat pada gambar tersebut, hasil pada awal mendapatkan nilai loss sebesar 2,57% dan akurasi sebesar 99,05% yang dicapai pada data training, sedangkan pada data validasi, nilai loss sebesar 6,01% dan akurasi sebesar 98,22%. Proses pelatihan ini berlanjut hingga akhir, hingga mendapatkan nilai loss sebesar 1,16% dan akurasi sebesar 99,43% dicapai pada data training, sedangkan pada data validasi, nilai loss sebesar 07,48% dan akurasi sebesar 98,00% yang dapat dilihat pada tabel 4.52.

Tabel 4.52 Hasil Fine Tuning MobileNetV2 50 Epoch 2 Kelas (CLAHE – White Balance)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
40/80	37ms	0.1432	0.9471	0.1121	0.9711	1.0000e-04
41/80	24ms	0.0611	0.9805	0.0884	0.9778	1.0000e-04
42/80	23ms	0.0563	0.9800	0.0558	0.9733	1.0000e-04
43/80	21ms	0.0529	0.9800	0.0866	0.9689	1.0000e-04
44/80	21ms	0.0486	0.9829	0.0918	0.9711	1.0000e-04
45/80	21ms	0.0681	0.9757	0.1758	0.9556	1.0000e-04
46/80	21ms	0.0524	0.9829	0.0727	0.9778	1.0000e-04
47/80	21ms	0.0523	0.9824	0.2015	0.9600	1.0000e-04
48/80	22ms	0.0567	0.9805	0.1089	0.9667	1.0000e-04
49/80	21ms	0.0532	0.9800	0.0619	0.9756	1.0000e-04
50/80	22ms	0.0304	0.9900	0.0559	0.9822	1.0000e-05
51/80	22ms	0.0257	0.9905	0.0601	0.9822	1.0000e-05
52/80	21ms	0.0193	0.9919	0.0831	0.9800	1.0000e-05
53/80	22ms	0.0165	0.9929	0.0785	0.9800	1.0000e-05
54/80	22ms	0.0151	0.9943	0.0654	0.9822	1.0000e-05
55/80	23ms	0.0153	0.9933	0.0774	0.9800	1.0000e-05
56/80	22ms	0.0152	0.9948	0.0786	0.9800	1.0000e-05
57/80	21ms	0.0154	0.9943	0.0760	0.9800	1.0000e-05
58/80	22ms	0.0167	0.9933	0.0963	0.9800	1.0000e-05
59/80	22ms	0.0126	0.9957	0.0864	0.9800	2.0000e-06
60/80	22ms	0.0073	0.9976	0.0967	0.9822	2.0000e-06
61/80	22ms	0.0153	0.9952	0.0954	0.9800	2.0000e-06
62/80	21ms	0.0111	0.9957	0.0869	0.9800	2.0000e-06
63/80	21ms	0.0120	0.9957	0.0864	0.9800	2.0000e-06
64/80	21ms	0.0078	0.9986	0.0955	0.9800	2.0000e-06
65/80	21ms	0.0092	0.9967	0.0942	0.9800	2.0000e-06
66/80	21ms	0.0147	0.9943	0.0945	0.9800	2.0000e-06
67/80	22ms	0.0094	0.9952	0.0955	0.9800	2.0000e-06
68/80	21ms	0.0098	0.9952	0.0951	0.9800	2.0000e-06
69/80	21ms	0.0089	0.9962	0.0982	0.9800	2.0000e-06
70/80	21ms	0.0078	0.9971	0.0946	0.9800	2.0000e-06
71/80	21ms	0.0138	0.9943	0.0693	0.9800	2.0000e-06
72/80	21ms	0.0122	0.9952	0.0960	0.9800	2.0000e-06
73/80	22ms	0.0139	0.9948	0.0825	0.9800	2.0000e-06
74/80	21ms	0.0125	0.9948	0.0897	0.9822	2.0000e-06

Tabel 4.52 Hasil Fine Tuning MobileNetV2 50 Epoch 2 Kelas (CLAHE – White Balance) (Lanjutan)

Epoch	Time /step	Training		Validation		Learning Rate
		Loss	Accuracy	Loss	Accuracy	
75/80	21ms	0.0104	0.9957	0.0808	0.9800	2.0000e-06
76/80	21ms	0.0155	0.9952	0.0807	0.9800	2.0000e-06
77/80	21ms	0.0146	0.9957	0.0869	0.9822	2.0000e-06
78/80	21ms	0.0123	0.9962	0.0790	0.9800	2.0000e-06
79/80	22ms	0.0146	0.9938	0.0946	0.9822	2.0000e-06
80/80	21ms	0.0116	0.9943	0.0748	0.9800	2.0000e-06

4.4. Analisa dan Pembahasan

Setelah menyelesaikan preprocessing data dan pelatihan pada dua arsitektur dan dua kelas, dilakukan pengujian menggunakan data uji untuk mengevaluasi akurasi dan performa dari masing-masing model. Pada tahap ini, beberapa parameter diatur dengan cara yang sama, yaitu menggunakan rasio dataset 70:15:15, di mana 70% dataset digunakan untuk pelatihan, 15% dataset digunakan sebagai data validasi, dan 15% data digunakan untuk evaluasi pada arsitektur yang telah dibangun. Model menggunakan optimizer Adam dengan learning rate 0,001, dan dilakukan pelatihan dengan 10, 20, dan 50 epoch. Hasil pengujian model klasifikasi skenario dapat dilihat dalam Tabel 4.53.

Tabel 4.53 Hasil Penelitian

ResNet50						
Jumlah Kelas	Epoch	Time /step	Akurasi	Precision	Recall	F1-Score
2	10	33ms	95,33%	95,33%	95,33%	95,33%
2	20	34ms	97,10%	97,11%	97,10%	97,11%
2	50	33ms	96,52%	96,52%	96,52%	96,52%
3	10	34ms	80,29%	80,26%	78,25%	78,28%
3	20	33ms	81,52%	81,62%	80,97%	80,97%
3	50	33ms	83,14%	82,86%	82,95%	83,11%

Tabel 4.53 Hasil Penelitian (Lanjutan)

ResNet50						
Jumlah Kelas	Epoch	Time /step	Akurasi	Precision	Recall	F1-Score
2	10	33ms	95,33%	95,33%	95,33%	95,33%
2	20	34ms	97,10%	97,11%	97,10%	97,11%
2	50	33ms	96,52%	96,52%	96,52%	96,52%
3	10	34ms	80,29%	80,26%	78,25%	78,28%
3	20	33ms	81,52%	81,62%	80,97%	80,97%
3	50	33ms	83,14%	82,86%	82,95%	83,11%
MobileNetV2						
Jumlah Kelas	Epoch	Time /step	Akurasi	Precision	Recall	F1-Score
2	10	22ms	92,48%	92,34%	92,34%	92,87%
2	20	21ms	93,42%	93,42%	93,42%	93,42%
2	50	21ms	93,76%	93,76%	93,76%	93,76%
3	10	23ms	91,90%	91,68%	91,79%	91,82%
3	20	22ms	95,54%	95,45%	95,54%	95,54%
3	50	21ms	95,51%	95,51%	95,35%	95,50%
ResNet50 – CLAHE – White Balance						
Jumlah Kelas	Epoch	Time /step	Akurasi	Precision	Recall	F1-Score
2	10	42ms	88,71%	89,08%	88,63%	88,71%
2	20	43ms	89,56%	90,56%	89,56%	89,55%
2	50	42ms	91,62%	91,12%	91,12%	91,12%
3	10	34ms	66,51%	65,94%	67,08%	65,79%
3	20	35ms	73,19%	74,11%	73,19%	73,19%
3	50	35ms	71,30%	72,10%	71,30%	71,30%
MobileNetV2 – CLAHE – White Balance						
Jumlah Kelas	Epoch	Time /step	Akurasi	Precision	Recall	F1-Score
2	10	22ms	99,57%	99,35%	99,31%	99,35%
2	20	22ms	99,67%	99,54%	99,34%	99,54%
2	50	21ms	99,76%	99,35%	99,35%	99,35%
3	10	23ms	95,90%	95,64%	94,66%	95,66%
3	20	21ms	92,51%	92,17%	92,17%	92,11%
3	50	23ms	91,17%	91,05%	91,05%	91,05%

Pada tabel 4.53 dapat dilihat hasil eksperimen yang dilakukan pada setiap skema ResNet50 dan MobileNetV2 memiliki performa yang baik dalam tugas klasifikasi citra. Pada dataset dengan 2 kelas, kedua model menunjukkan kemampuan yang sangat baik dalam mengklasifikasikan citra, dengan tingkat akurasi yang tinggi di atas 91,17% sampai 99,76 untuk ResNet50 dan di sekitar 88,71% hingga 97,10% untuk MobileNetV2 setelah beberapa epoch. Selain itu, presisi, recall, dan F1-Score juga mencapai tingkat yang tinggi, menunjukkan kemampuan model dalam mengenali dan membedakan kelas dengan baik.

Ketika diterapkan pada dataset dengan 3 kelas, kedua model tetap menunjukkan performa yang baik meskipun tingkat akurasi sedikit lebih rendah dibandingkan dengan dataset 2 kelas. Model ResNet50 mencapai akurasi sekitar 66,76% hingga 83,14% setelah beberapa epoch, sementara MobileNetV2 mencapai akurasi sekitar 91,17% hingga 95,90%. Meskipun terdapat perbedaan dalam performa, keduanya tetap memiliki kemampuan yang memadai dalam mengklasifikasikan citra multi-kelas.

Penggunaan teknik CLAHE dan white balance pada kedua model memberikan peningkatan performa dalam beberapa kasus, terutama pada dataset dengan 2 kelas. Penerapan teknik ini menghasilkan peningkatan akurasi yang signifikan, dengan MobileNetV2 bahkan mencapai akurasi 99% setelah 10 epoch. Namun, penggunaan teknik tersebut tidak memberikan peningkatan yang signifikan pada dataset dengan 3 kelas, menunjukkan bahwa pengaruh teknik pengolahan citra tersebut tergantung pada karakteristik dataset yang digunakan.

Selain itu, perlu diperhatikan bahwa MobileNetV2 memiliki keunggulan dalam waktu komputasi time per-step yang lebih cepat dibandingkan dengan ResNet50, menjadikannya pilihan yang lebih efisien dalam pengolahan citra dengan skala besar.

Dalam keseluruhan, baik model ResNet50 maupun MobileNetV2 adalah pilihan yang baik untuk tugas klasifikasi citra CXR, tergantung pada kebutuhan spesifik. Keputusan pemilihan model harus mempertimbangkan jumlah kelas, waktu komputasi, dan teknik pengolahan citra yang relevan.

4.5. Perbandingan dengan Penelitian Sebelumnya

Tabel 4.54 menunjukkan perbandingan hasil antara penelitian ini dan penelitian sebelumnya yang menggunakan dataset yang sama. Dalam penelitian ini, penggunaan kombinasi white balance, CLAHE, dan arsitektur CNN menghasilkan tingkat akurasi yang lebih tinggi dibandingkan dengan metode-metode lain yang telah diajukan oleh peneliti sebelumnya. Pendekatan ini berhasil meningkatkan akurasi pelatihan pada dataset yang sama.

Tabel 4.54 Perbandingan Metode Dengan Penelitian Sebelumnya

Kelas	Penelitian	Pre-Processing	Metode	Akurasi
2 Kelas	(D. S. Kermany et al., 2018)	–	CNN	92,8%
	(Ayan & Unver, 2019)	Augmentation Data	VGG16	87%
	(Saraiva et al., 2019)	–	CNN	95,30%
	(Stephen et al., 2019)	Augmentation Data	CNN	93,73%
	(Luján-García et al., 2020)	Cost-Sensitive Learning	Xception	97%
	(Chhikara et al., 2020)	DIP algorithms	InceptionV3	90,1%
	(Mittal et al., 2020)	Augmentation Data	ECC	95,90%
	(Liang & Zheng, 2020)	–	ResNet	90,5%
	(Labhane et al., 2020)	Augmentation Data	InceptionV3	98%
	(Çınar et al., 2020)	–	ResNet50	97,22%

Tabel 4.54 Perbandingan Metode Dengan Penelitian Sebelumnya (Lanjutan)

Kelas	Penelitian	Pre-Processing	Metode	Akurasi
2 Kelas	(Muhammad et al., 2021)	Augmentation Data, CLAHE	InceptionV3	97,19%
	(Dey et al., 2021)	–	VGG19	97,94%
	(Kundu et al., 2021)	–	GoogLeNet, ResNet-18, DenseNet-121	98,81%
	(El Asnaoui et al., 2021)	Augmentation Data, CLAHE	Resnet50	96.61%
	(Sanchez et al., 2022)	–	CX-DaGAN	97,78%
	(Trivedi & Gupta, 2022)	Augmentation Data	MobileNet	97.34%
	Penelitian yang diajukan	**Augmentation Data, CLAHE, White Balance**	**MobileNetV2**	**99,76%**
3 Kelas	(Yadav & Jadhav, 2019)	Augmentation Data	VGG16	90,7%
	(Ayan et al., 2022)	Augmentation Data	PNet	90,71%
	Penelitian yang diajukan	**Augmentation Data, CLAHE, White Balance**	**MobileNetV2**	**95,90%**

Pada tabel 4.54 dapat dilihat untuk dataset dengan klasifikasi 2 kelas, berbagai metode dan teknik yang diterapkan memberikan hasil yang baik dalam mengklasifikasikan citra CXR. Beberapa metode yang mencapai tingkat akurasi yang tinggi termasuk penggunaan CNN, Augmentation Data, Cost-Sensitive Learning, dan DIP algorithms. Model-model seperti VGG16, Xception, ResNet50, InceptionV3, dan GoogLeNet juga menunjukkan performa yang baik dengan akurasi di atas 90% hingga mencapai hampir 99%. Selain itu, teknik seperti ECC (Ensemble of convolutions with capsules), CLAHE, dan penggunaan CX-DaGAN juga memberikan peningkatan performa yang signifikan.

Namun, ketika kelas pneumonia diperluas menjadi 3 kelas, tingkat akurasi umumnya sedikit lebih rendah dibandingkan dengan dataset 2 kelas. Meskipun demikian, penggunaan Augmentation Data dan model seperti VGG16, PNet, dan MobileNetV2 tetap memberikan hasil yang baik dengan akurasi di atas 90% hingga

mencapai 95%. Hal ini menunjukkan bahwa meskipun dataset menjadi lebih kompleks, model-model tersebut tetap mampu mengklasifikasikan citra multi-kelas dengan baik.

Selain itu, penggunaan teknik Augmentation Data, CLAHE, dan White Balance memberikan peningkatan performa yang signifikan, terutama pada dataset dengan 2 kelas. Teknik-teknik tersebut berhasil meningkatkan akurasi model MobileNetV2 hingga mencapai 99,76%. Namun, pada dataset dengan 3 kelas, pengaruh teknik pengolahan citra ini tampaknya tidak sebesar pada dataset 2 kelas.

Oleh karena itu, pemilihan teknik pengolahan citra harus mempertimbangkan karakteristik dataset yang digunakan. Keputusan pemilihan model harus mempertimbangkan jumlah kelas, waktu komputasi, dan teknik pengolahan citra yang relevan.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan pengujian dan analisis yang dilakukan terhadap dua arsitektur model yang berbeda untuk klasifikasi penyakit pneumonia pada gambar CXR, dapat diambil kesimpulan sebagai berikut:

1. Dari hasil eksperimen, dapat disimpulkan bahwa kedua model, yaitu MobileNetV2 dan ResNet50, memiliki performa yang baik dalam tugas klasifikasi penyakit pneumonia pada dataset CXR. Namun, MobileNetV2 menunjukkan performa yang sedikit lebih baik dibandingkan dengan ResNet50. MobileNetV2 mampu mencapai tingkat akurasi yang tinggi 93,76% untuk dataset dengan 2 kelas, sementara ResNet50 mencapai akurasi 97,10%. Pada dataset dengan 3 kelas, performa keduanya tetap baik dengan akurasi 83,14% untuk ResNet50 dan 95,54% untuk MobileNetV2.
2. Terdapat perbedaan dalam waktu komputasi time per-step antara ResNet50 dan MobileNetV2. MobileNetV2 memiliki keunggulan dalam waktu komputasi per-step yang lebih cepat dibandingkan dengan ResNet50. Ini menjadikan MobileNetV2 sebagai pilihan yang lebih efisien dalam pengolahan citra dengan skala besar dengan kecepatan yang konstan antara 21ms hingga 23ms. Jika efisiensi waktu menjadi faktor penting, MobileNetV2 dapat menjadi pilihan yang lebih disukai.

3. Penggunaan teknik CLAHE dan white balance memberikan peningkatan performa dalam beberapa kasus, terutama pada dataset dengan 2 kelas. Penerapan teknik ini menghasilkan peningkatan akurasi yang signifikan, dengan MobileNetV2 bahkan mencapai akurasi 99,57% setelah 10 epoch. Namun, penggunaan teknik tersebut tidak memberikan peningkatan yang signifikan pada dataset dengan 3 kelas, menunjukkan bahwa pengaruh teknik pengolahan citra tersebut tergantung pada karakteristik dataset yang digunakan. Dalam pemilihan teknik pengolahan citra, perlu dipertimbangkan karakteristik dataset dan apakah teknik tersebut memberikan peningkatan performa yang signifikan.

5.2. Saran

Berikut ini adalah beberapa rekomendasi yang bisa menjadi panduan dalam pengembangan penelitian ini, antara lain:

1. Melakukan analisis mendalam tentang mekanisme pengoptimalan dan adaptasi dari kedua algoritma tersebut.
2. Memperluas cakupan penelitian dengan menggunakan dataset yang lebih besar dan beragam untuk menguji performa arsitektur dalam klasifikasi pneumonia.
3. Meneliti faktor-faktor lain yang dapat memengaruhi time consumption.
4. Melakukan analisis perbandingan kinerja dalam hal penggunaan sumber daya komputasi dan efisiensi memori antara kedua arsitektur untuk memahami trade-off yang mungkin terjadi.
5. Melakukan eksperimen dengan kombinasi teknik dan mengidentifikasi strategi yang paling efektif dalam meningkatkan performa model.

DAFTAR PUSTAKA

PUSTAKA BUKU

Ballard, D. H., & Brown, C. M. (1982). *Computer Vision* (1st Edition). Prentice Hall.

Elgendy, M. (2020). Deep Learning for Vision Systems. Manning Publications.

Eng, P., & Cheah, F.-K. (2005). Interpreting Chest X-Rays. Cambridge University Press. https://doi.org/10.1017/CBO9780511545368

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. http://www.deeplearningbook.org

Ke, Q., Liu, J., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2018). Chapter 5 - Computer Vision for Human–Machine Interaction. In M. Leo & G. M. Farinella (Eds.), Computer Vision for Assistive Healthcare (pp. 127–145). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-813445-0.00005-8

Mostafa, S., & Wu, F.-X. (2021). Chapter 3 - Diagnosis of autism spectrum disorder with convolutional autoencoder and structural MRI images. In A. S. El-Baz & J. S. Suri (Eds.), Neural Engineering Techniques for Autism Spectrum Disorder (pp. 23–38). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-822822-7.00003-X

Szeliski, R. (2011). Computer Vision. Springer London. https://doi.org/https://dsoi.org/10.1007/978-1-84882-935-0

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Abraham, B., & Nair, M. S. (2020). Computer-aided detection of COVID-19 from X-ray images using multi-CNN and Bayesnet classifier. *Biocybernetics and Biomedical Engineering*, *40*(4), 1436–1445. https://doi.org/https://doi.org/10.1016/j.bbe.2020.08.005

Agarap, A. F. (2018). Deep Learning using Rectified Linear Units (ReLU). *CoRR*, *abs/1803.08375*. http://arxiv.org/abs/1803.08375

Ahsan, F., Rahmawati, N. Y., & Alditia, F. N. (2020). *Lawan Virus Corona: Studi Nutrisi untuk Kekebalan Tubuh* (B. Santoso, Ed.). Airlangga University Press.

Ayan, E., Karabulut, B., & Ünver, H. M. (2022). Diagnosis of Pediatric Pneumonia with Ensemble of Deep Convolutional Neural Networks in Chest X-Ray Images. *Arabian Journal for Science and Engineering*, *47*(2), 2123–2139. https://doi.org/10.1007/s13369-021-06127-z

Ayan, E., & Unver, H. M. (2019). Diagnosis of Pneumonia from Chest X-Ray Images Using Deep Learning. *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, 1–5. https://doi.org/10.1109/EBBT.2019.8741582

Balachandran, G. (2014). *Interpretation of Chest X-Ray: An Illustrated Companion* (1st ed., Vol. 1). Jaypee Brothers Medical Publishers.

Ballard, D. H., & Brown, C. M. (1982). *Computer Vision* (1st Edition). Prentice Hall.

Bera, S., & Shrivastava, V. K. (2020). Effect of pooling strategy on convolutional neural network for classification of hyperspectral remote sensing images. *IET Image Processing*, *14*(3), 480–486. https://doi.org/10.1049/iet-ipr.2019.0561

Chhikara, P., Singh, P., Gupta, P., & Bhatia, T. (2020). *Deep Convolutional Neural Network with Transfer Learning for Detecting Pneumonia on Chest X-Rays* (pp. 155–168). https://doi.org/10.1007/978-981-15-0339-9_13

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.

Çınar, A., Yıldırım, M., & Eroğlu, Y. (2020). Classification of Pneumonia Cell Images Using Improved ResNet50 Model. *International Information and Engineering Technology Association*, *38*(1). https://doi.org/https://doi.org/10.18280/ts.380117

Collins, J., & Stern, E. J. (2015). *Chest Radiology: The Essentials (Essentials Series) - Third Edition* (J. Collins, Ed.; 3rd ed., Vol. 1). Wolters Kluwer.

Corne, J., & Pointon, K. (2010). *Chest X-Ray Made Easy- Third Edition* (3rd ed., Vol. 1). Churchill Livingstone Elsevier.

Daniel, Cenggoro, T. W., & Pardamean, B. (2023). A systematic literature review of machine learning application in COVID-19 medical image classification. *Procedia Computer Science*, *216*, 749–756. https://doi.org/https://doi.org/10.1016/j.procs.2022.12.192

Daoud, M. I., Alrahahleh, Y., Abdel-Rahman, S., Alsaify, B. A., & Alazrai, R. (2021). COVID-19 Diagnosis in Chest X-ray Images by Combining Pre-trained CNN Models with Flat and Hierarchical Classification Approaches. *2021 12th International Conference on Information and Communication Systems (ICICS)*, 330–335. https://doi.org/10.1109/ICICS52457.2021.9464532

Desai, M., & Shah, M. (2021). An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN). *Clinical EHealth*, *4*, 1–11. https://doi.org/10.1016/j.ceh.2020.11.002

Dey, N., Zhang, Y.-D., Rajinikanth, V., Pugalenthi, R., & Raja, N. S. M. (2021). Customized VGG19 Architecture for Pneumonia Detection in Chest X-Rays. *Pattern Recognition Letters*, *143*, 67–74. https://doi.org/10.1016/j.patrec.2020.12.010

Dilshad, S., Singh, N., Atif, M., Hanif, A., Yaqub, N., Farooq, W. A., Ahmad, H., Chu, Y., & Masood, M. T. (2021). Automated image classification of chest X-rays of COVID-19 using deep transfer learning. *Results in Physics*, *28*, 104529. https://doi.org/https://doi.org/10.1016/j.rinp.2021.104529

El Asnaoui, K., Chawki, Y., & Idri, A. (2021). *Automated Methods for Detection and Classification Pneumonia Based on X-Ray Images Using Deep Learning* (pp. 257–284). https://doi.org/10.1007/978-3-030-74575-2_14

Elgendy, M. (2020). *Deep Learning for Vision Systems*. Manning Publications.

Emin Sahin, M. (2022). Deep learning-based approach for detecting COVID-19 in chest X-rays. *Biomedical Signal Processing and Control*, *78*, 103977. https://doi.org/https://doi.org/10.1016/j.bspc.2022.103977

Eng, P., & Cheah, F.-K. (2005). *Interpreting Chest X-Rays*. Cambridge University Press. https://doi.org/10.1017/CBO9780511545368

Estomihi, J. (2019). *Klasifikasi Penyakit Pneumonia dari Citra X-Ray Menggunakan Backpropagation* [Undergraduate Papers]. Universitas Sumatera Utara.

Firmansyah, I., & Hayadi, B. H. (2022). Komparasi Fungsi Aktivasi Relu Dan Tanh Pada Multilayer Perceptron. *JIKO (Jurnal Informatika Dan Komputer)*, *6*(2), 200. https://doi.org/10.26798/jiko.v6i2.600

Ghose, P., Uddin, Md. A., Acharjee, U. K., & Sharmin, S. (2022). Deep viewing for the identification of Covid-19 infection status from chest X-Ray image using CNN based architecture. *Intelligent Systems with Applications*, *16*, 200130. https://doi.org/https://doi.org/10.1016/j.iswa.2022.200130

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *Computing Research Repository (CoRR)*. https://doi.org/https://doi.org/10.48550/arXiv.1512.03385

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Ke, Q., Liu, J., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2018). Chapter 5 - Computer Vision for Human–Machine Interaction. In M. Leo & G. M. Farinella (Eds.), *Computer Vision for Assistive Healthcare* (pp. 127–145). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-813445-0.00005-8

Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M. Y. L., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., … Zhang, K. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, *172*(5), 1122-1131.e9. https://doi.org/10.1016/j.cell.2018.02.010

Kermany, D., Zhang, K., & Goldbaum, M. (2018). *Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification*. https://doi.org/10.17632/rscbjbr9sj.2

Khan, M. M. R., Sakib, S., Siddique, Md. A. B., Chowdhury, M., Hossain, Z., Aziz, A., & Yasmin, N. (2020). Automatic Detection of COVID-19 Disease in Chest X-Ray Images using Deep Neural Networks. *2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC)*, 1–6. https://doi.org/10.1109/R10-HTC49770.2020.9357034

Kolonne, S., Kumarasinghe, H., Fernando, C., & Meedeniya, D. (2021, September). MobileNetV2 Based Chest X-Rays Classification. *2021 International Conference on Decision Aid Sciences and Application (DASA)*. https://doi.org/10.1109/DASA53625.2021.9682248

Kumar Sethy, P., Kumari Behera, S., Kumar Ratha, P., & Biswas, P. (2020). Detection of Coronavirus Disease (COVID-19) based on Deep Features and Support Vector Machine. *International Journal of Mathematical Engineering and Management Sciences*, *5*(4), 643–651. https://doi.org/https://doi.org/10.33889/IJMEMS.2020.5.4.052

Kundu, R., Das, R., Geem, Z. W., Han, G.-T., & Sarkar, R. (2021). Pneumonia detection in chest X-ray images using an ensemble of deep learning models. *PLOS ONE*, *16*(9), e0256630. https://doi.org/10.1371/journal.pone.0256630

Labhane, G., Pansare, R., Maheshwari, S., Tiwari, R., & Shukla, A. (2020). Detection of Pediatric Pneumonia from Chest X-Ray Images using CNN and Transfer Learning. *2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE)*, 85–92. https://doi.org/10.1109/ICETCE48199.2020.9091755

LeCun, Y., & Bengio, Y. (1995). *Convolutional networks for images, speech, and time series* (M.A. Arbib, Ed.; Vol. 3361). MIT Press.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Liang, G., & Zheng, L. (2020). A transfer learning method with deep residual network for pediatric pneumonia diagnosis. *Computer Methods and Programs in Biomedicine*, *187*, 104964. https://doi.org/10.1016/j.cmpb.2019.06.023

Ligueran, R. J. S. D., Santos, M. L. C. D., Tinio, Dr. R. S., & Valencia, E. H. (2022). Applied Computer Vision on 2-Dimensional Lung X-Ray Images for Assisted Medical Diagnosis of Pneumonia. *International Journal of Computing Sciences Research*, *6*. https://doi.org/10.25147/ijcsr.2017.001.1.98

Liu, S., Cai, T., Tang, X., Zhang, Y., & Wang, C. (2022). COVID-19 diagnosis via chest X-ray image classification based on multiscale class residual attention. *Computers in Biology and Medicine*, *149*, 106065. https://doi.org/https://doi.org/10.1016/j.compbiomed.2022.106065

Luján-García, J., Yáñez-Márquez, C., Villuendas-Rey, Y., & Camacho-Nieto, O. (2020). A Transfer Learning Method for Pneumonia Classification and Visualization. *Applied Sciences*, *10*(8), 2908. https://doi.org/10.3390/app10082908

Lu, S., Wang, S.-H., & Zhang, Y.-D. (2020). Detecting pathological brain via ResNet and randomized neural networks. *Heliyon*, *6*(12), e05625. https://doi.org/https://doi.org/10.1016/j.heliyon.2020.e05625

Mittal, A., Kumar, D., Mittal, M., Saba, T., Abunadi, I., Rehman, A., & Roy, S. (2020). Detecting Pneumonia Using Convolutions and Dynamic Capsule Routing for Chest X-ray Images. *Sensors*, *20*(4), 1068. https://doi.org/10.3390/s20041068

Mostafa, S., & Wu, F.-X. (2021). Chapter 3 - Diagnosis of autism spectrum disorder with convolutional autoencoder and structural MRI images. In A. S. El-Baz & J. S. Suri (Eds.), *Neural Engineering Techniques for Autism Spectrum Disorder* (pp. 23–38). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-822822-7.00003-X

Muhammad, Y., Alshehri, M. D., Alenazy, W. M., Vinh Hoang, T., & Alturki, R. (2021). Identification of Pneumonia Disease Applying an Intelligent Computational Framework Based on Deep Learning and Machine Learning Techniques. *Mobile Information Systems*, *2021*, 1–20. https://doi.org/10.1155/2021/9989237

Naveen, P., & Diwan, B. (2021). Pre-trained VGG-16 with CNN Architecture to classify X-Rays images into Normal or Pneumonia. *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 102–105. https://doi.org/10.1109/ESCI50559.2021.9396997

Nazir, S., Dickson, D. M., & Akram, M. U. (2023). Survey of explainable artificial intelligence techniques for biomedical imaging with deep neural networks. *Computers in Biology and Medicine*, *156*, 106668. https://doi.org/https://doi.org/10.1016/j.compbiomed.2023.106668

Niederman, M.S, Craven, D.E, & MJ, B. (2005). Guidelines for the management of adults with hospital-acquired, ventilator-associated, and healthcare-associated pneumonia. *American Journal of Respiratory and Critical Care Medicine*, *171*(4), 388–416.

Reed, J. C. (2011). *Chest Radiology: Plain Film Patterns and Differential Diagnoses* (6th ed., Vol. 1). Elsevier Mosby.

Riskesdas. (2019). *Hasil Utama Riskesdas 2018*. https://www.litbang.kemkes.go.id/hasil-utama-riskesdas-2018/

Rizal Makarim, F. (2022). *X-Ray*. Halodoc. https://www.halodoc.com/kesehatan/x-ray

Rosenbusch, G., & Eekelen, A. de K. (2019). *Wilhelm Conrad Röntgen: The Birth of Radiology* (1st ed.). Springer Cham. https://doi.org/https://doi.org/10.1007/978-3-319-97661-7

Roy, S., Tyagi, M., Bansal, V., & Jain, V. (2022). SVD-CLAHE boosting and balanced loss function for Covid-19 detection from an imbalanced Chest X-ray dataset. *Computers in Biology and Medicine*, 106092. https://doi.org/https://doi.org/10.1016/j.compbiomed.2022.106092

Safira, S., Djohan, S., & Nurjanana, N. (2019). Pengaruh pengeluaran pemerintah pada bidang infrastruktur pendidikan dan kesehatan terhadap pertumbuhan ekonomi di provinsi kalimantan timur. *Jurnal Ekonomi Manajemen Dan Akutansi*, *21*(2).

Sanchez, K., Hinojosa, C., Arguello, H., Kouame, D., Meyrignac, O., & Basarab, A. (2022). CX-DaGAN: Domain Adaptation for Pneumonia Diagnosis on a

Small Chest X-Ray Dataset. *IEEE Transactions on Medical Imaging*, *41*(11), 3278–3288. https://doi.org/10.1109/TMI.2022.3182168

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Saraiva, A., Ferreira, N., Lopes de Sousa, L., Costa, N., Sousa, J., Santos, D., Valente, A., & Soares, S. (2019). Classification of Images of Childhood Pneumonia using Convolutional Neural Networks. *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies*, 112–119. https://doi.org/10.5220/0007404301120119

Shadab, S. A., Ansari, M. A., Singh, N., Verma, A., Tripathi, P., & Mehrotra, R. (2022). Chapter 15 - Detection of cancer from histopathology medical image data using ML with CNN ResNet-50 architecture. In R. Agrawal, M. A. Ansari, R. S. Anand, S. Sneha, & R. Mehrotra (Eds.), *Computational Intelligence in Healthcare Applications* (pp. 237–254). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-323-99031-8.00007-7

Siddhartha, M., & Santra, A. (2020). COVIDLite: A depth-wise separable deep neural network with white balance and CLAHE for detection of COVID-19. *Journal of Computer Methods and Programs*.

Stephen, O., Sain, M., Maduh, U. J., & Jeong, D.-U. (2019). An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare. *Journal of Healthcare Engineering*, *2019*, 1–7. https://doi.org/10.1155/2019/4180949

Szeliski, R. (2011). *Computer Vision*. Springer London. https://doi.org/https://doi.org/10.1007/978-1-84882-935-0

Torres, A., Cilloniz, C., Niederman, M. S., Menéndez, R., Chalmers, J. D., Wunderink, R. G., & van der Poll, T. (2021). Pneumonia. *Nature Reviews Disease Primers*, *7*(1), 25. https://doi.org/10.1038/s41572-021-00259-0

Trivedi, M., & Gupta, A. (2022). A lightweight deep learning architecture for the automatic detection of pneumonia using chest X-ray images. *Multimedia Tools and Applications*, *81*(4), 5515–5536. https://doi.org/10.1007/s11042-021-11807-x

Tukbekova, B. T., Kizatova, S. T., Zhanpeissova, A. A., Dyussenova, S. B., Serikova, G. B., Isaeva, A. A., Tlegenova, K. S., & Kiryanova, T. A. (2019). Causes of delayed immunization with pneumococcal vaccine and aetiological patterns of pneumonia in young children. *Revista Latinoamericana de Hipertensión*, *14*(3), 337–345. https://www.redalyc.org/articulo.oa?id=170263176021

Umri, B. K., Utami, E., & Kurniawan, M. P. (2021). Comparative Analysis of CLAHE and AHE on Application of CNN Algorithm in the Detection of Covid-19 Patients. *2021 4th International Conference on Information and Communications Technology (ICOIACT)*, 203–208. https://doi.org/10.1109/ICOIACT53268.2021.9563980

Venkatesan, R., & Li, B. (2018). Convolutional neural networks in visual computing : a concise guide. In *Convolutional neural networks in visual computing : a concise guide*. CRC Press.

Wu, J.-N. (2016). Compression of fully-connected layer in neural network by Kronecker product. *2016 Eighth International Conference on Advanced Computational Intelligence (ICACI)*, 173–179. https://doi.org/10.1109/ICACI.2016.7449822

Yadav, S. S., & Jadhav, S. M. (2019). Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big Data*, *6*(1), 113. https://doi.org/10.1186/s40537-019-0276-2

Yingge, H., Ali, I., & Lee, K.-Y. (2020). Deep Neural Networks on Chip - A Survey. *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 589–592. https://doi.org/10.1109/BigComp48618.2020.00016

Yopento, J., Ernawati, & Coastera Farady, F. (2022). Identifikasi Pneumonia Pada Citra X-Ray Paru-Paru Menggunakan Metode Convolutional Neural Network (Cnn) Berdasarkan Ekstraksi Fitur Sobel. *Jurnal Rekursif*, *10*(1). https://doi.org/https://doi.org/10.33369/rekursif.v10i1.17247

PUSTAKA LAPORAN PENELITIAN

Riskesdas. (2019). *Hasil Utama Riskesdas 2018*. https://www.litbang.kemkes.go.id/hasil-utama-riskesdas-2018/

PUSTAKA ELEKTRONIK

Rizal Makarim, F. (2022). *X-Ray*. Halodoc. https://www.halodoc.com/kesehatan/x-ray

Kermany, D., Zhang, K., & Goldbaum, M. (2018). Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification. https://doi.org/10.17632/rscbjbr9sj.2

LAMPIRAN