IMPLEMENTASI ARSITEKTUR RESNET50 DAN RESNET101 PADA SISTEM KEHADIRAN BERBASIS *FACE RECOGNITION*

(Skripsi)

Oleh

ERVAN CHODRY
NPM 2017051001

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024

IMPLEMENTASI ARSITEKTUR RESNET50 DAN RESNET101 PADA SISTEM KEHADIRAN BERBASIS *FACE RECOGNITION*

Oleh

ERVAN CHODRY

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER

Pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024

ABSTRAK

IMPLEMENTASI ARSITEKTUR RESNET50 DAN RESNET101 PADA SISTEM KEHADIRAN BERBASIS *FACE RECOGNITION*

Oleh

ERVAN CHODRY

Salah satu bidang yang berkembang pesat setelah munculnya teknologi *machine learning* adalah teknologi *face recognition*. *Convolutional Neural Network* (CNN) merupakan salah satu metode yang sangat berperan dalam perkembangan teknologi *face recognition*. CNN memiliki berbagai jenis arsitektur yang dapat digunakan pada teknologi *face recognition*. Penelitian ini bertujuan untuk mengetahui akurasi dari model CNN dengan arsitektur ResNet50 dan ResNet101 yang diimplementasikan pada teknologi *face recognition*. Penelitian ini dilaksanakan di Jurusan Ilmu Komputer, Universitas Lampung, pada bulan Oktober - Desember 2023. *Dataset* yang digunakan adalah 400 gambar wajah dari 20 mahasiswa. *Dataset* tersebut dibagi menjadi 40 data uji dan 360 data latih. Data latih yang telah dipisahkan kemudian melalui proses *preprocessing* dan augmentasi hingga jumlahnya meningkat menjadi total 14400 gambar. Data latih tersebut kemudian dibagi kembali menjadi 11520 data latih dan 2880 data validasi. Hasil validasi pada saat *training* model menunjukkan model ResNet50 mendapat akurasi terbaik dengan nilai 96% dan model ResNet101 mendapat akurasi terbaik sebesar 85%. Namun hasil pengujian menunjukkan kedua model hanya mendapat akurasi sebesar 60%. Hal ini diduga karena jumlah *dataset* yang digunakan masih sangat sedikit untuk jumlah *class* yang cukup banyak.

Kata kunci: *Convolutional Neural Network* (CNN), Pengenalan Wajah, ResNet50, ResNet101

ABSTRACT

IMPLEMENTATION OF RESNET50 AND RESNET101 ARCHITECTURES IN A FACE RECOGNITION-BASED ATTENDANCE SYSTEM

By

ERVAN CHODRY

The burgeoning field of face recognition technology, driven by advancements in machine learning, has witnessed the widespread implementation of Convolutional Neural Network (CNN) methods. CNN, equipped with various architectural designs, plays a pivotal role in shaping the landscape of face recognition technology. This study, conducted at the Computer Science Department, University of Lampung, from October to December 2023, focuses on assessing the accuracy of CNN models employing ResNet50 and ResNet101 architectures in the realm of face recognition technology. A dataset consisting of 400 facial images from 20 individuals was utilized, with 40 images designated for testing and 360 for training. Preceding the application of the dataset, preprocessing and augmentation techniques were employed, resulting in an augmented dataset of 14,400 images. Further subdivision of this augmented dataset yielded 11,520 training images and 2,880 validation images. During the training phase, ResNet50 demonstrated the highest accuracy at 96%, while ResNet101 exhibited commendable performance with an accuracy of 85%. However, the subsequent testing phase revealed that both models achieved an accuracy of approximately 60%. This discrepancy is attributed to the relatively limited size of the dataset, posing challenges in accommodating the diverse array of facial features present in the broader population. The findings underscore the importance of adequately sized datasets for the effective deployment of CNN models in face recognition technology.

Keywords: Convolutional Neural Network (CNN), Face Recognition, ResNet50, ResNet101

Judul Skripsi	:	**IMPLEMENTASI ARSITEKTUR RESNET50 DAN RESNET101 PADA SISTEM KEHADIRAN BERBASIS FACE RECOGNITION**
Nama Mahasiswa	:	Ervan Chodry
Nomor Pokok Mahasiswa	:	2017051001
Program Studi	:	S1 Ilmu Komputer
Jurusan	:	Ilmu Komputer
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Anie Rose Irawati, S.T., M.Cs.
NIP. 19791031 200604 2 002

Dewi Asiah Shofiana, S.Komp., M.Kom.
NIP. 19950929 202012 2 030

2. Ketua Jurusan Ilmu Komputer

Didik Kurniawan, S.Si., M.T.
NIP. 19800419 200501 1 004

MENGESAHKAN

1. Tim Penguji

Ketua : **Anie Rose Irawati, S.T., M.Cs.** .....

Sekretaris : **Dewi Asiah Shofiana, S.Komp., M.Kom.** .....

Penguji utama : **Didik Kurniawan, S.Si., M.T.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 19711001 200501 1 002

Tanggal Lulus Ujian Skripsi: **18 Januari 2024**

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ervan Chodry

NPM : 2017051001

Menyatakan bahwa skripsi saya yang berjudul **"Implementasi Arsitektur Resnet50 dan Resnet101 pada Sistem Kehadiran Berbasis Face Recognition"** merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 6 Februari 2024

Ervan Chodry
NPM. 2017051001

RIWAYAT HIDUP

Penulis dilahirkan di Bandar Lampung pada tanggal 11 Juni 2002 sebagai anak pertama dari tiga bersaudara, dari bapak Sariyanto dan Ibu Tri Wahyuni. Pendidikan yang telah ditempuh oleh penulis diantaranya, menyelesaikan pendidikan dasar di SD Negeri 3 Rejomulyo Jati Agung Lampung Selatan pada tahun 2014. Penulis menyelesaikan pendidikan menengah pertama di SMP Negeri 2 Natar Lampung Selatan pada tahun 2017. Penulis melanjutkan pendidikan menengah atas di SMA Negeri 4 Metro pada tahun 2020. Perjalan pendidikan penulis dilanjutkan dengan terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur Seleksi Nilai Masuk Perguruan Tinggi (SNMPTN) pada tahun 2020.

Selama menjadi mahasiswa, penulis aktif mengikuti beberapa kegiatan antara lain:

1. Anggota UKM Koperasi Mahasiswa Universitas Lampung.
2. Asisten Praktikum mata kuliah Logika, Pemrograman Terstruktur, Pemrograman Desktop, Pemrograman Berorientasi Objek, dan Pemrograman Web Lanjut.
3. Pada tahun 2022 menjadi pengurus di UKM Koperasi Mahasiswa Universitas Lampung.
4. Pada tahun 2023 mendapatkan kesempatan mengikuti kegiatan magang di PT. Sentra Vidya Utama sebagai *Fullstack Web Developer*.

MOTO

“Kita tidak dapat memprediksi masa depan, tetapi kita dapat menciptakannya”

(Bill Gates)

PERSEMBAHAN

Kupersembahkan karya ini kepada:

Kedua Orang Tuaku Tercinta

Yang senantiasa memberikan yang terbaik, dan melantunkan do'a yang selalu Menyertaiku. Kuucapkan pula terimakasih sebesar-besarnya karena telah mendidik dan membesarkanku dengan cara yang dipenuhi kasih sayang, dukungan, dan pengorbanan yang belum bisa terbalaskan.

SANWACANA

Puji syukur atas segala rahmat Allah SWT yang diberikan kepada penulis sehingga dapat menyelesaikan skripsi berjudul "Implementasi Arsitektur ResNet50 dan ResNet101 pada Sistem Kehadiran Berbasis *Face Recognition*". Tidak lupa shalawat dan salam senantiasa dicurahkan kepada Nabi Muhammad SAW, suri teladan yang telah menunjukkan jalan yang benar kepada seluruh umatnya.

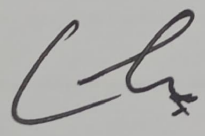
Pada kesempatan ini penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan skripsi ini, yaitu:

1. Allah SWT, yang telah memberikan nikmat, rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan lancar hingga akhir.
2. Kedua orang tua yang senantiasa menyemangati, mendukung, mendoakan, dan membantu segala hal yang tidak terhitung nilainya.
3. Ibu Anie Rose Irawati, S.T. M.Cs. selaku dosen pembimbing atas kesediaannya untuk memberikan bimbingan, arahan, kritik, serta sarannya selama proses penyelesaian skripsi.
4. Ibu Dewi Asiah Shofiana, S.Komp., M.Kom. selaku dosen pembimbing yang selalu membantu memberikan saran dan masukan selama proses penyelesaian skripsi.
5. Bapak Didik Kurniawan, S.Si., M.T. selaku ketua Jurusan Ilmu Komputer Universitas Lampung serta dosen pembahas yang telah memberikan kritik, saran serta masukan kepada penulis.
6. Bapak dan Ibu Dosen Jurusan Ilmu Komputer Universitas Lampung yang telah memberikan ilmu, pengetahuan serta pengalamannya selama penulis menjadi mahasiswa.

7. Ibu Ade Nora Maela selaku staf administrasi Jurusan Ilmu Komputer Universitas Lampung yang telah banyak membantu proses administrasi dalam proses penyelesaian skripsi.
8. Seluruh staf Jurusan Ilmu Komputer yang selalu membantu penulis mulai dari awal hingga akhir masa kuliah.
9. Arib Yusron Hamdani, Regita Rose Prameswari, M. Hanif Pratama, Adiwijaya Nusantara, dan Irvandra Dwidya Agsatra sebagai rekan penelitian yang selalu dapat menjadi tempat bertukar pikiran.
10. M. Daffa Putra Wibowo selaku teman yang telah bersama-sama menyelesaikan tugas akhir mulai dari awal hingga akhir.
11. Teman-teman "Nyelow" yang telah menemani dari awal masa perkuliahan.
12. Semua pihak yang turut membantu penulis dalam penyusunan laporan skripsi ini.

Penulis menyadari bahwa masih banyak kekurangan dalam penulisan skripsi ini, akan tetapi semoga skripsi ini dapat membawa manfaat dan keberkahan bagi perkembangan ilmu pengetahuan terutama bagi semua civitas Ilmu Komputer Universitas Lampung.

Bandar Lampung, 6 Februari 2024



Ervan Chodry
NPM. 2017051001

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

I. PENDAHULUAN

1.1. Latar Belakang

Machine learning saat ini telah mengalami perkembangan yang sangat pesat. Teknologi ini telah digunakan di berbagai bidang untuk membantu meringankan pekerjaan manusia. Mulai dari pengolahan citra, pemrosesan bahasa alami, pengenalan suara, hingga analisis data. Banyak algoritma dan teknik *machine learning* yang terus dikembangkan untuk memperbaiki kinerja model dan meningkatkan kemampuan model dalam menangani data yang semakin kompleks.

Salah satu bidang yang berkembang pesat bersama dengan perkembangan teknologi *machine learning* adalah *face recognition*. Beberapa tahun terakhir, banyak teknologi yang mulai menerapkan *face recognition* mulai dari teknologi keamanan, identifikasi, hingga pengalaman pengguna. Berbagai teknik *machine learning* seperti *deep learning* dan *Convolutional Neural Network* (CNN) dimanfaatkan dan diimplementasikan untuk meningkatkan keakuratan dan efisiensi teknologi *face recognition*. CNN merupakan salah satu jenis *deep learning* yang digunakan untuk mengolah data dua dimensi dan banyak diimplementasikan pada data gambar (Nugroho dkk., 2020).

Pemanfaatan CNN dalam teknologi *face recognition* membuat teknologi ini dapat berkembang dengan sangat pesat. CNN dapat membantu meningkatkan keakuratan dan efisiensi sistem *face recognition*. CNN memiliki tingkat akurasi yang lebih tinggi apabila dibandingkan dengan metode tradisional seperti LBPH (Wijaya dkk., 2023). CNN memungkinkan model untuk mempelajari fitur-fitur kompleks pada pola wajah manusia

yang sulit diidentifikasi oleh metode konvensional. Selain itu, teknik *deep learning* yang terintegrasi dalam CNN dapat meningkatkan kemampuan model dalam memahami dan mengenali variasi wajah manusia yang sangat kompleks.

Arsitektur CNN yang dapat digunakan dalam *face recognition* adalah *Residual Network* (ResNet). Arsitektur ini merupakan arsitektur CNN yang diperkenalkan oleh Kaiming He dan timnya pada tahun 2015. Arsitektur ResNet mampu mendapatkan hasil akurasi yang lebih tinggi namun dengan *training error* yang lebih rendah apabila dibandingkan dengan arsitektur CNN lain (He dkk., 2016). ResNet50 dan ResNet101 merupakan dua dari beberapa arsitektur ResNet. Perbedaan dari kedua arsitektur tersebut terletak pada jumlah *layer* yang digunakan. Sesuai namanya ResNet50 menggunakan 50 *layer* dan ResNet101 menggunakan 101 *layer*.

Salah satu sistem yang dapat mengimplementasikan teknologi *face recognition* adalah sistem presensi. Penelitian terkait penerapan *face recognition* pada sistem presensi pernah dilakukan oleh Ramdhon & Febriya (2021). Penelitian tersebut menggunakan kamera pengawas untuk mendeteksi wajah mahasiswa yang hadir di ruang kelas. Penelitian tersebut menggunakan algoritma *Local Binary Pattern Histogram* (LBPH) untuk mendeteksi wajah dari *input* yang diberikan. Hasil pengujian menunjukkan bahwa sistem yang dibuat mendapatkan akurasi sebesar 86,85% ketika mengenali wajah mahasiswa yang berada pada ruang kelas (Ramdhon & Febriya, 2021).

Berdasarkan pemaparan tersebut karya tulis ilmiah berjudul "**Implementasi Arsitektur ResNet50 dan ResNet101 Pada Sistem Kehadiran Berbasis *Face Recognition***" ini akan membahas pengembangan model *machine learning* yang menggunakan arsitektur ResNet50 dan ResNet101 yang akan diintegrasikan dengan sebuah *web service*. Dengan mengintegrasikan teknologi berbasis *deep learning*, seperti ResNet50 dan ResNet101 diharapkan dapat membantu meningkatkan efisiensi dan efektifitas dalam proses pengenalan wajah. Dengan dibuatnya model dengan kedua arsitektur

ini diharapkan dapat digunakan tidak hanya pada sistem presensi namun juga pada berbagai bidang lain yang menggunakan *face recognition*. Selain itu, tingkat keamanan data juga diharapkan dapat meningkat setelah diintegrasikan dengan teknologi *face recognition*.

1.2. Rumusan Masalah

Berdasarkan pemaparan pada latar belakang maka rumusan masalah dalam penelitian ini adalah bagaimana mengimplementasikan arsitektur ResNet50 dan ResNet101 untuk digunakan pada sistem kehadiran berbasis *face recognition*.

1.3. Batasan Masalah

Adapun batasan masalah dalam penelitian ini yaitu sebagai berikut.

a. Penelitian ini berfokus pada pembuatan model dengan metode *Convolutional Neural Network* dan *Web Service* untuk menerima dan memprediksi foto yang dikirimkan melalui sistem presensi.
b. *Dataset* yang digunakan merupakan kumpulan gambar dengan format warna RGB yang memiliki format ekstensi *file* berupa (.jpg dan .png) yang terdiri dari beberapa foto mahasiswa yang dikelompokkan berdasarkan Nomor Pokok Mahasiswa (NPM).
c. Model yang dibuat mengimplementasikan arsitektur ResNet50 dan ResNet101.

1.4. Tujuan

Penelitian ini memiliki beberapa tujuan, yaitu:

a. Mengetahui akurasi model yang menggunakan arsitektur ResNet50 dan ResNet101.

b. Mengembangkan model dengan arsitektur ResNet50 dan ResNet101 yang dapat memprediksi mahasiswa berdasarkan foto wajah.
c. Mengembangkan *Web Services* yang dapat menerima foto wajah mahasiswa dan mengembalikan respon berupa prediksi mahasiswa yang ada pada foto tersebut.

1.5. Manfaat

Manfaat dari penelitian ini adalah sebagai berikut:

a. Membantu meningkatkan efisiensi dalam melakukan pencatatan kehadiran mahasiswa dalam perkuliahan.
b. Mempermudah dosen dalam membuat rekapitulasi kehadiran mahasiswa.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian yang pernah dilakukan dan berkaitan dengan pembuatan sistem presensi berbasis *face recognition* menggunakan arsitektur ResNet50 dan ResNet101 dapat dilihat pada Tabel 1.

Tabel 1. Penelitian terdahulu.

Penulis	Judul	Metode	Hasil
(Ramdhon & Febriya, 2021)	Penerapan *Face Recognition* pada Sistem Presensi	*Local Binary Pattern Histogram* (LBPH)	Sistem yang dibuat berhasil mendapat akurasi pengujian sebesar 86,85% ketika mengenali wajah mahasiswa yang hadir di ruang kelas.
(Wajdi & Sugiantara, 2018)	Pemanfaatan Teknik Pengenalan Wajah Berbasis OpenCV untuk Sistem Informasi Pencatatan Kehadiran Dosen	*Design Science*	Penelitian tersebut berhasil membuat sebuah *prototype* sistem kehadiran yang dapat mengenali wajah secara *real-time* menggunakan *library* OpenCV.
(Elpeltagy & Sallam, 2021)	*Automatic Prediction of Covid-19 from Chest Images using Modified ResNet50*	*Convolutional Neural Network* (CNN)	Penelitian tersebut berhasil menghasilkan arsitektur baru dengan basis ResNet50 yang disebut Covid-ResNet53 yang digunakan untuk

Penulis	Judul	Metode	Hasil
			mendeteksi virus Covid-19 melalui hasil *X-ray* dan *CT-Scan*. Model yang dibuat dengan arsitektur tersebut memiliki akurasi 10,4% lebih baik pada foto *CT-Scan* dan 8,5% lebih baik pada foto *X-ray* dibandingkan model yang menggunakan arsitektur VGG19.
(Oktaviana dkk., 2021)	Klasifikasi Penyakit Padi Berdasarkan Citra Daun Menggunakan Model Terlatih ResNet101	*Convolutional Neural Network* (CNN)	Penelitian tersebut berhasil membuat model dengan arsitektur ResNet101 untuk mengklasifikasikan penyakit padi berdasarkan citra daun. Model yang dibuat mendapatkan akurasi 100% dengan nilai *validation loss* 5,61%.

Penelitian terkait penerapan teknologi pengenalan wajah pada sistem kehadiran pernah dilakukan oleh (Ramdhon & Febriya, 2021) dengan judul penelitian "Penerapan *Face Recognition* pada Sistem Presensi". Penelitian ini menggunakan *library* OpenCV dengan metode *Local Binary Pattern Histogram* (LBPH) untuk mengenali wajah. *Local Binary Pattern Histogram* (LBPH) merupakan sebuah kombinasi antara algoritma *Local Binary Pattern* (LBP) dan *Histogram of Oriented Gradients* (HOG) dan merupakan penyempurnaan performa dari algoritma LBP dalam proses *face recognition* (Ramdhon & Febriya, 2021). Penelitian ini menggunakan kamera pengawas sebagai perangkat yang akan mendeteksi wajah mahasiswa di ruang kelas, kemudian data mahasiswa yang terdeteksi akan disimpan pada *database*. Pengujian pada penelitian ini dilakukan atas 30 responden yang terdiri dari 18 laki-laki dan 12 perempuan yang berada pada

sebuah ruang kelas. Pengujian dilakukan sebanyak 5 iterasi dengan kondisi yang berbeda-beda. Hasil yang didapatkan dari pengujian yang dilakukan sistem mendapatkan akurasi sebesar 86,85% dan kegagalan sebesar 13,15%.

Penelitian terkait penerapan teknologi pengenalan wajah pada sistem kehadiran juga pernah dilakukan oleh (Wajdi & Sugiantara, 2018) dengan judul penelitian "Pemanfaatan Teknik Pengenalan Wajah Berbasis Opencv untuk Sistem Informasi Pencatatan Kehadiran Dosen". Penelitian ini menggunakan model Opencv untuk melakukan pengenalan wajah. Model Opencv merupakan *library* dengan lisensi *open source*, yang ditujukan untuk pengolahan gambar secara *real-time* (Wajdi & Sugiantara, 2018). Untuk dapat mendeteksi dan mengenali wajah, penelitian ini menerapkan algoritma *Eigenface*. Algoritma *Eigenface* merupakan salah satu algoritma pengenalan pola yang berdasarkan pada *Principle Component Analysis* (PCA). Dalam algoritma ini informasi unik pada pola wajah disimpan dengan merepresentasikan *eigen-vector* menjadi sebuah matriks berukuran besar (Wajdi & Sugiantara, 2018). Dalam penelitian tersebut berhasil dibuat sebuah sistem *prototype* yang dapat mengenali wajah yang muncul pada kamera secara *real-time*. Hasil yang didapat pada pengujian sistem dengan metode *Technology Acceptance Model* (TAM) menunjukkan bahwa nilai tiap indikator pengujian mendapat rata-rata nilai lebih dari 4,00 yang mana nilai antara 3,68-5,00 mendapat interpretasi "Baik".

Penelitian terkait implementasi arsitektur ResNet50 pernah dilakukan oleh (Elpeltagy & Sallam, 2021) dengan judul "*Automatic Prediction of Covid-19 from Chest Images Using Modified Resnet50*". Penelitian ini bertujuan untuk membangun model dengan basis arsitektur ResNet50 yang dapat mendeteksi pasien yang terpapar virus Covid-19 berdasarkan foto hasil *X-ray* dan *CT-scan* pada bagian dada atau paru-parunya. *Dataset* yang digunakan terdiri dari 5500 foto *X-ray* pasien Non-Covid, 4045 foto *X-ray* pasien Covid, 2628 foto *CT-Scan* pasien Non-Covid, dan 5427 foto *CT-Scan* pasien Covid. Dalam penelitian ini berhasil dibuat sebuah arsitektur baru dengan basis ResNet50 yang diberi nama arsitektur Covid-ResNet53.

Arsitektur ini dibuat khusus untuk mendeteksi pasien yang terpapar virus Covid-19 berdasarkan foto hasil *X-ray* atau *CT-Scan*. Setelah dilakukan pengujian, arsitektur Covid-ResNet53 memiliki akurasi yang lebih baik dibandingkan arsitektur VGG19 sebesar 10,4% pada foto *CT-Scan* dan 8,5% pada foto *X-Ray.*

Penelitian terkait implementasi arsitektur ResNet101 telah dilakukan oleh (Oktaviana dkk., 2021) dengan judul penelitian "Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih ResNet101". Penelitian ini bertujuan untuk membuat model *machine learning* dengan menerapkan arsitektur ResNet101 ditambah dengan arsitektur rancangan yang diusulkan untuk klasifikasi penyakit pada tanaman padi berdasarkan foto kondisi daun padi. *Dataset* yang digunakan pada penelitian ini diambil dari situs repositori terbuka bernama *UCI Machine Learning Repository* dengan judul data *Rice Leaf Diseases Dataset*, yang terdiri dari tiga kelas penyakit padi yaitu *bacterial leaf blight*, *brown spot*, dan *leaf smut* yang masing-masing berjumlah 40 foto sehingga total foto yang digunakan berjumlah 120 foto. *Dataset* tersebut kemudian dibagi menjadi data *training* dan validasi dengan jumlah 108 data *training* dan 12 data validasi. Setelah dilakukan *training*, dihasilkan sebuah model yang memiliki nilai akurasi validasi sebesar 100% dan validasi *loss* sebesar 5,61%. Berdasarkan hasil evaluasi yang dilakukan menggunakan *confusion matrix* menunjukkan bahwa model mendapatkan akurasi 1 pada setiap kelas yang merepresentasikan bahwa model memiliki performa yang bagus (Oktaviana dkk., 2021).

2.1.Pengenalan Wajah

Pengenalan wajah merupakan suatu teknologi biometrik yang dapat mengenali wajah seseorang dari sebuah foto atau video berdasarkan ciri-ciri wajahnya (Detila dkk., 2019). Untuk dapat mengenali wajah seseorang diperlukan data ciri-ciri wajah yang akan dideteksi untuk disimpan atau dipelajari pola wajahnya oleh sebuah model. Model tersebut akan

mengenali wajah dengan membandingkan ciri-ciri wajah yang ada di sebuah foto atau video dengan data wajah yang telah disimpan atau dipelajari sebelumnya.

Pengenalan wajah telah diterapkan pada berbagai sektor. Dalam sektor keamanan pengenalan wajah digunakan untuk mendeteksi apabila terdapat orang tak dikenal yang muncul pada kamera keamanan. Dalam sektor administrasi pengenalan wajah telah banyak digunakan sebagai metode dalam pencatatan kehadiran. Teknologi pengenalan wajah dianggap dapat menjadi alternatif yang sangat relevan untuk tujuan keamanan sistem dan pengenalan sidik jari dan retina mata (Wajdi & Sugiantara, 2018).

2.2. *Machine Learning*

Machine learning merupakan bidang studi yang membuat mesin atau komputer untuk dapat belajar tanpa perlu diprogram secara eksplisit (Samuel, 1959). *Machine learning* dapat mempelajari pola dalam *dataset* dan menggunakannya untuk membuat prediksi atau keputusan yang akurat di masa mendatang. Secara sederhana, *machine learning* adalah kemampuan komputer untuk belajar dari pengalaman. Komputer akan mempelajari *dataset* yang diberikan untuk membuat keputusan atau prediksi di masa mendatang. Pola yang dipelajari oleh komputer dari *dataset* akan diubah menjadi sebuah algoritma tertentu yang kemudian akan disimpan menjadi sebuah model *machine learning*. Model ini nantinya yang akan digunakan untuk membuat prediksi atau keputusan di masa mendatang.

Dalam *machine learning* tingkat akurasi dalam membuat keputusan atau prediksi bergantung pada kualitas *dataset* yang dipelajari. Hal ini karena komputer akan belajar berdasarkan *dataset* yang diberikan. Semakin baik kualitas *dataset* yang diberikan maka akan semakin akurat prediksi atau keputusan yang dihasilkan. Sebaliknya semakin kurang kualitas *dataset* yang diberikan maka semakin kurang juga prediksi atau keputusan yang

dihasilkan. Secara umum *machine learning* dapat dibagi menjadi dua jenis yaitu *supervised learning* dan *unsupervised learning*.

a. *Supervised Learning*

Supervised Learning merupakan metode *machine learning* menggunakan data yang telah diberi label sesuai dengan hasil yang diharapkan. Komputer akan mempelajari pola yang terdapat pada data berlabel untuk kemudian digunakan memprediksi data baru yang belum memiliki label. Secara umum metode ini terbagi lagi menjadi Regresi dan Klasifikasi. Regresi merupakan metode *supervised learning* yang menggunakan data yang berkelanjutan, seperti data harga rumah setiap bulan. Klasifikasi merupakan metode *supervised learning* yang menggunakan data dalam bentuk kategori, seperti data gambar hewan yang terdiri dari kucing dan anjing. Contoh algoritma dalam *supervised learning* adalah *Decision Tree*, KNN (*K-Nearest Neighbors*), *Logistic Regression,* SVM, dan *Naïve Baiyes*.

b. *Unsupervised Learning*

Unsupervised learning merupakan metode *machine learning* yang menggunakan data yang tidak diberi label sebagai *dataset*. Komputer akan mempelajari pola atau tren tersembunyi dari *dataset* yang diberikan. Kemudian pola tersembunyi tersebut akan digunakan untuk mengelompokkan data berdasarkan kemiripan polanya. Contoh algoritma *unsupervised learning* adalah *K-Means Clustering* dan *Hierarchical clustering.*

2.3. *Neural Network*

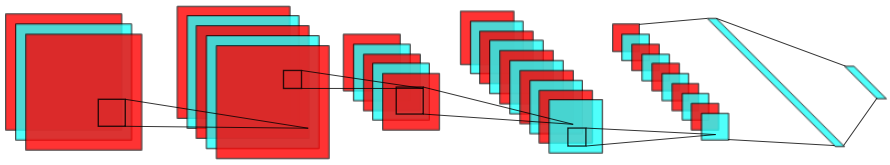
Neural Network merupakan model *machine learning* yang terinspirasi dari sistem saraf manusia dengan menerapkan *Multilayer Perceptron* (MLP). MLP merupakan jenis jaringan saraf tiruan yang terdiri dari satu atau lebih

hidden layer. Komponen utama dari MLP terdiri dari *input layer*, *hidden layer*, dan *output layer*. MLP yang tidak memiliki *hidden layer* biasa disebut *Perceptron.* Setiap *layer* pada MLP memiliki satu atau lebih unit pemrosesan yang disebut *neuron*. *Neuron* berisi fungsi aktivasi yang akan menerima *input* dari *neuron* lain, kemudian menghasilkan *output* yang dikirim ke *neuron* lain.

Setiap *neuron* yang ada pada suatu *neural network* terhubung satu sama lain untuk menghasilkan *output* tertentu. Untuk menghasilkan *output*, *neural network* akan menerima *input layer* yang kemudian akan diproses melalui *neuron-neuron* yang ada pada *hidden layer*. Proses pada setiap *neuron* dilakukan dengan menghitung nilai *input* yang diterima dengan *weight* tertentu. Jumlah *hidden layer* pada suatu *neural network* yang biasa disebut *network depth* dapat mencapai tak terhingga. *Neural network* yang memiliki *hidden layer* yang cukup banyak biasa disebut *Deep Neural Network* atau lebih dikenal dengan *Deep Learning* (Carleo dkk., 2019).

2.4. *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) merupakan salah satu jenis *deep learning* yang digunakan untuk mengolah data dua dimensi dan banyak diimplementasikan pada data citra (Nugroho dkk., 2020). CNN biasa digunakan untuk menganalisis gambar visual, mendeteksi dan mengenali objek pada gambar berupa vektor berdimensi tinggi. Seperti pada *neural network* lainnya CNN terdiri dari banyak *hidden layer* yang terdiri dari *neuron-neuron* yang saling terhubung antar *layer*.



Gambar 1. Ilustrasi *Convolutional Neural Network*.

CNN merupakan algoritma yang bekerja secara hierarki, artinya *output* yang dihasilkan pada sebuah *layer* akan digunakan sebagai *input* pada *layer* berikutnya (Setyawan dkk., 2023). Proses pada *Convolutional Neural Network* diilustrasikan seperti pada Gambar 1. Ilustrasi tersebut menggambarkan sebuah model CNN yang memiliki beberapa *layer* seperti *convolution layer*, *max pooling*, *flatten layer*, dan *fully connected layer*. Ilustrasi tersebut juga menunjukkan bahwa setiap *output* dari setiap *layer* saling terhubung karena digunakan sebagai *input* pada *layer* berikutnya. Secara umum CNN terdiri dari beberapa *layer* seperti *input layer, convolution layer, pooling layer, flatten layer,* dan *fully connected layer*.

a. *Convolution Layer*

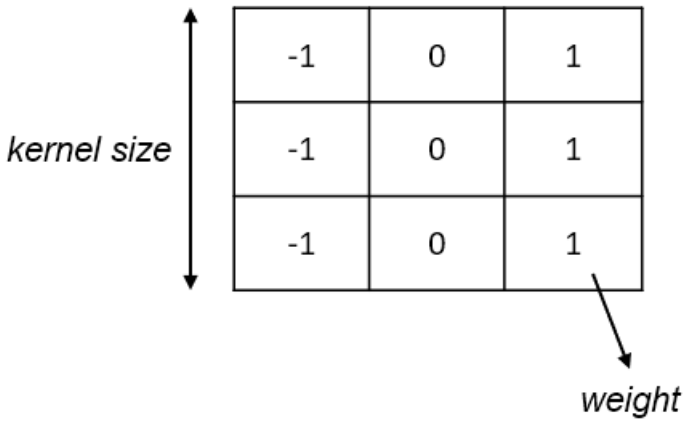
Convolution layer digunakan untuk mengekstrak fitur dari *input* dengan operasi konvolusi. Operasi konvolusi merupakan operasi yang dilakukan dengan menjumlahkan hasil perkalian elemen-*wise* dari matriks filter *F* dengan ukuran *nxn* dengan sebuah area dari *input*, operasi ini dilakukan secara terus menerus hingga semua area pada *input layer* telah tercakup (Setyawan dkk., 2023).

Convolution layer memiliki beberapa komponen utama yaitu *kernel*, *stride,* dan *padding* (Albawi dkk., 2017).

1) *Kernel*

Kernel merupakan gabungan beberapa *input* yang saling berdekatan dalam ukuran tertentu. Ukuran *kernel* akan menentukan jumlah

input berdekatan yang akan digabungkan untuk menjadi sebuah *output*. Untuk mendapatkan sebuah *output* dari suatu *kernel, input kernel* akan diproses dengan cara dikalikan dengan sebuah matriks yang disebut filter. Setiap elemen pada filter merepresentasikan nilai *weight* dari setiap elemen *kernel*. Filter dapat digunakan untuk mengekstrak fitur pada gambar. Ilustrasi kernel pada CNN dapat dilihat pada Gambar 2.



Gambar 2. Ilustrasi *kernel* pada CNN.

2) *Stride*

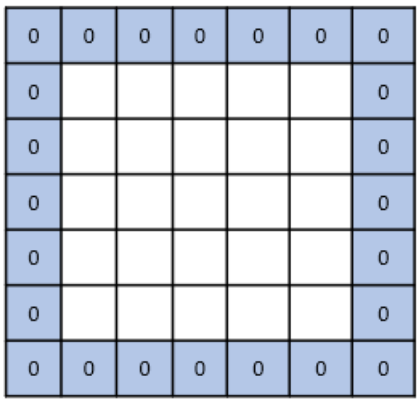
Stride merupakan jarak perpindahan posisi *kernel* dari posisi sebelumnya. Perpindahan *kernel* dilakukan secara horizontal ke arah kanan hingga mencapai posisi *input* terakhir yang berada di tepi kanan gambar. *Kernel* kemudian akan berpindah ke bawah sebanyak nilai *stride* kemudian akan kembali berpindah ke kanan hingga mencapai titik akhir gambar. Perpindahan kernel yang dilakukan pada setiap iterasi membuat beberapa *kernel* akan saling tumpang tindih dan menghasilkan *output* yang tidak efektif. Oleh karena itu perpindahan *kernel* dapat diatur untuk menentukan seberapa banyak *input* yang saling tumpang tindih menggunakan *stride*. Ilustrasi

perpindahan kernel dengan *stride* yang bernilai 1 dapat dilihat pada Gambar 3.

Gambar 3. Ilustrasi *stride* pada CNN.

3) *Padding*

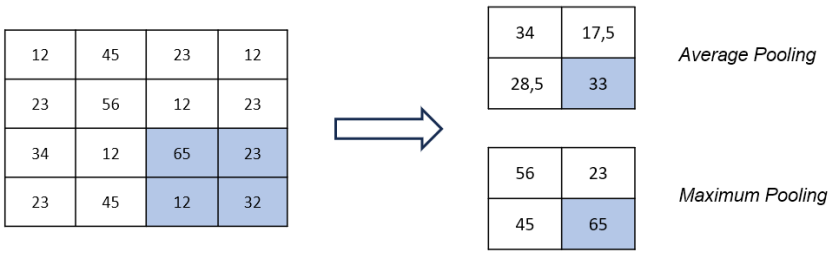
Padding merupakan nilai tambahan yang diletakkan pada setiap sisi *input layer*. Fitur yang terletak pada sisi tepi gambar seringkali tidak terbaca dan mengakibatkan hilangnya beberapa informasi dari *input layer*. Hal ini karena ukuran *layer* akan terus berkurang ketika melewati proses konvolusi. Untuk mengatasi hal tersebut sebuah nilai tambahan dapat ditambahkan pada setiap sisi *input* untuk mencegah hilangnya informasi yang terletak pada tepi gambar, sehingga ukuran *output* yang terbentuk akan sama dengan ukuran *input layer*. Nilai yang ditambahkan pada setiap sisi biasanya bernilai 0. Ilustrasi sebuah *input* yang menggunakan *padding* sebanyak 1 dapat dilihat pada Gambar 4.



Gambar 4. Ilustrasi *padding* pada sebuah *input*.

b. *Pooling Layer*

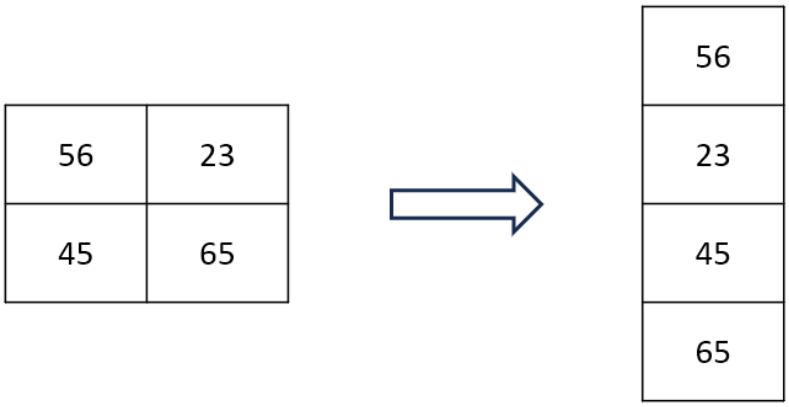
Pooling layer adalah *layer* yang digunakan untuk mengurangi *spatial resolution* dari *feature map* yang dihasilkan *convolution layer*. Operasi untuk mengurangi *spatial resolution* disebut dengan *spatial pooling*. Operasi *spatial pooling* yang paling umum adalah *average pooling* dan *maximum pooling*. *Maximum pooling* dilakukan dengan membagi *feature map* menjadi matriks berukuran $m \times m$, kemudian diambil nilai maksimum yang ada pada tiap matriks bagian sebagai fitur akhir. *Average pooling* akan mengambil nilai rata-rata dari seluruh elemen yang ada pada matriks bagian sebagai fitur akhir (Setyawan dkk., 2023). Ilustrasi *layer* yang dihasilkan pada *pooling layer* dapat dilihat pada Gambar 5.



Gambar 5. Ilustrasi proses pada *pooling layer.*

c. *Flatten*

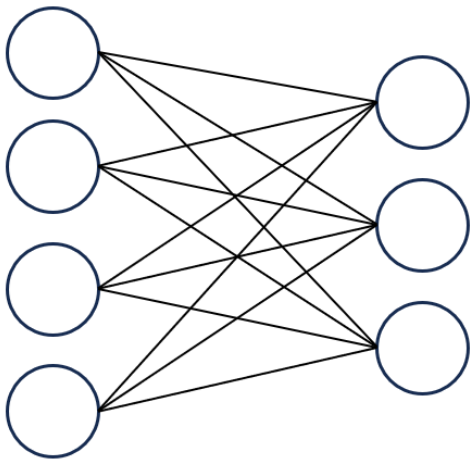
Flatten merupakan proses merubah *feature map* hasil konvolusi yang berbentuk matriks menjadi vektor (Setyawan dkk., 2023). Proses *flatten* dilakukan setelah *convolutional layer* dan *pooling layer*. Proses ini dilakukan agar *layer* yang dihasilkan pada proses sebelumnya dapat diproses pada *fully connected layer*. Proses *flatten* diilustrasikan pada Gambar 6.



Gambar 6. Ilustrasi proses *flatten.*

d. *Fully Connected Layer*

Fully Connected Layer merupakan *layer* pada CNN yang semua *neuron* pada *layer* saling dihubungkan dengan semua *neuron* yang ada pada *fully connected layer* berikutnya (Setyawan dkk., 2023). Ilustrasi dari *fully connected layer* dapat dilihat pada Gambar 7. *Layer* ini berfungsi untuk membuat prediksi atau keputusan akhir berdasarkan proses yang telah dilalui pada *hidden layer*. Prediksi atau keputusan yang diambil pada *layer* ini dilakukan dengan mengaplikasikan *weights* yang telah ditentukan pada setiap elemen yang ada pada vektor *input*.



Gambar 7. Ilustrasi *fully connected layer.*

2.5. *Residual Network* (ResNet)

Residual network (ResNet) merupakan arsitektur *neural network* yang diperkenalkan oleh Kaiming He dkk, pada tahun 2015. ResNet didesain untuk dapat mengatasi masalah *vanishing gradients* pada sebuah *neural network* yang memiliki *hidden layer* yang sangat banyak. Masalah ini muncul ketika ukuran gradasi menjadi sangat kecil ketika melewati jaringan, sehingga menjadi sulit untuk melakukan *training*. ResNet mampu menyelesaikan masalah ini dengan memperkenalkan *skip connections*. *Skip connections* memungkinkan gradasi untuk melewati *layer* yang ada tanpa perlu melewati proses yang ada pada *layer* tersebut. Dengan adanya *skip connections* memungkinkan ResNet untuk mempelajari *residual mapping* antara *input* dan *output* pada setiap *layer*. Pada percobaan menggunakan *dataset* ImageNet menunjukkan bahwa arsitektur ResNet yang memiliki semakin banyak *layer* akan mendapatkan *training error* yang lebih rendah, serta akurasi yang lebih tinggi (He dkk., 2016).

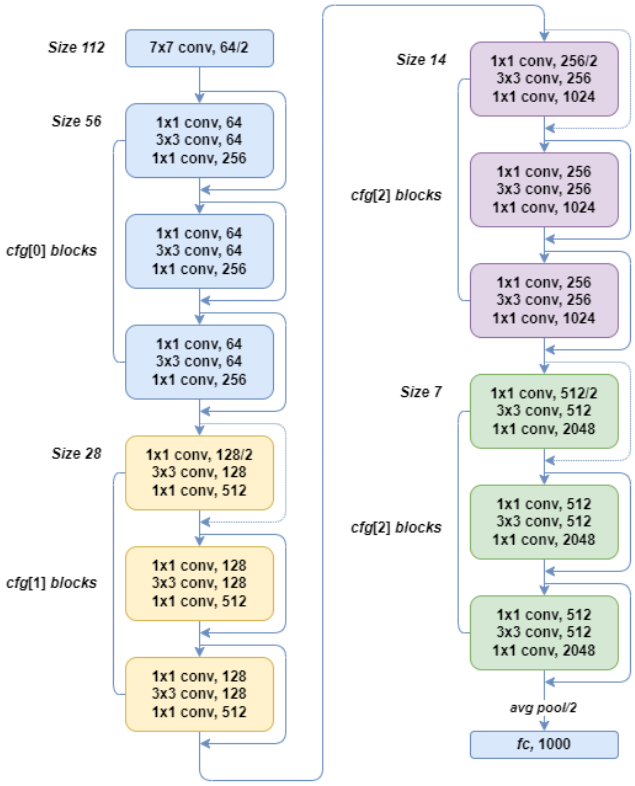
Dalam percobaannya, He dkk menguji beberapa jenis arsitektur ResNet pada *dataset* ImageNet, CIFAR-10, dan MS-COCO. Mereka menemukan terdapat beberapa jenis arsitektur ResNet yang cukup optimal untuk

digunakan seperti ResNet18, ResNet34, ResNet50, ResNet101, dan ResNet152. Perbedaan pada setiap jenis arsitektur ResNet terletak pada jumlah *layer* yang digunakan. Penelitian ini akan berfokus untuk mengimplementasikan arsitektur ResNet50 dan ResNet101. Tabel 2 merupakan rincian lapisan yang digunakan pada arsitektur ResNet50 dan ResNet101 berdasarkan penelitian yang dilakukan oleh He dkk.

Tabel 2. Rincian *layer* arsitektur ResNet50 dan ResNet101 (He dkk., 2016).

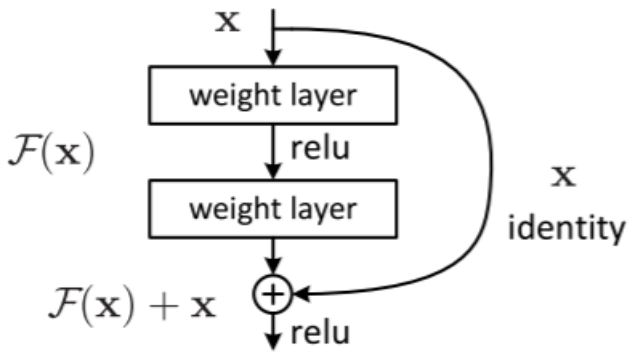
Layer	**Ukuran *output***	**ResNet50**	**ResNet101**
conv1	112×112	7x7, 64, *stride* 2	
		3x3 *max pool*, *stride* 2	
conv2_x	56×56	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4_x	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$
conv5_x	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	*Average pool*, 1000-d fc, *softmax*	
FLOPs		3.8×10^9	7.6×10^9

Pada jaringan ResNet terdapat modifikasi pada blok *shortcut connection*. Modifikasi tersebut dilakukan dengan menumpuk hasil dari *shortcut connection* dengan tiga *layer* konvolusi yang disebut *residual block* dengan ukuran dari masing-masing *layer* adalah 1x1, 3x3, dan 1x1. Modifikasi ini disebut *Deeper Bottleneck Architecture*, yang mana lapisan konvolusi terakhir berfungsi untuk mengurangi dan meningkatkan dimensi dengan meninggalkan lapisan konvolusi 3x3 dari *input/output* yang lebih kecil. Pada arsitektur ResNet 50, 101, dan 152 *layer* digunakan *parameter-free identity shortcut* (Niswati dkk., 2021).



Gambar 8. Arsitektur ResNet (Niswati dkk., 2021).

Arsitektur dari jaringan ResNet diilustrasikan pada Gambar 8. Struktur dasar dari arsitektur ResNet adalah penggunaan *residual block* dengan *shortcut connection* sebagai *layer* utama. Perbedaan antara arsitektur ResNet50 dan ResNet101 terdapat pada jumlah *residual block*. *Residual block* pada ResNet50 berjumlah 16 blok, sedangkan pada ResNet101 berjumlah 33 blok. Hal ini berpengaruh dengan kemampuan model dalam mengolah dan menampung relasi yang lebih kompleks. Semakin banyak *hidden layer* dalam sebuah model, maka semakin kompleks pula relasi yang dapat diolah dan ditampung (Pannakkong dkk., 2022). Struktur dari *residual block* diilustrasikan pada Gambar 9.



Gambar 9. *Residual block* (He dkk., 2016).

2.6. *Dataset*

Dataset merupakan kumpulan data dalam konteks tertentu yang digunakan untuk melatih serta menguji model *machine learning*. Untuk melatih sebuah model *machine learning*, memerlukan *dataset* yang memiliki kesamaan konteks seperti *dataset* foto penyakit padi, foto wajah, pergerakan harga saham, dan sebagainya. Salah satu faktor yang mempengaruhi akurasi dari sebuah model *machine learning* adalah kualitas dan variasi *dataset* yang digunakan. Seperti yang dilakukan oleh (Haryanto dkk., 2023), untuk meningkatkan kualitas *dataset* yang digunakan mereka melakukan *preprocessing* dengan membuang data yang tidak valid, tidak konsisten, dan data ganda. Selain itu (Masruroh dkk., 2023) juga melakukan *augmentasi* untuk meningkatkan variasi dari *dataset* yang mereka gunakan. *Augmentasi* adalah proses memperbanyak *dataset* dengan cara memberi variasi berdasarkan data yang sudah ada, seperti memperbesar atau memperkecil, memutar, mengubah warna foto, dan sebagainya.

2.7. *Preprocessing*

Preprocessing merupakan tahap yang digunakan untuk mempersiapkan data untuk dianalisis atau *modeling*. Dalam *machine learning*,

preprocessing menjadi sebuah tahap penting yang mengubah data mentah menjadi suatu format yang dapat digunakan oleh sebuah model. *Preprocessing* dapat melibatkan berbagai teknik seperti pembersihan data, normalisasi, *scaling*, *feature extraction*, dan *dimensionality reduction*. *Preprocessing* bertujuan untuk meningkatkan kualitas data dan untuk mempermudah model untuk mempelajari pola serta membuat prediksi yang lebih akurat. *Preprocessing* dapat mengurangi kompleksitas per-iterasi dan menghindari *update* yang berulang dari struktur data yang sama (Song dkk., 2021).

2.8. *Training Model*

Training merupakan proses model mempelajari pola dari *dataset* yang diberikan. Pada tahap *training* terdapat dua proses utama yaitu *feed-forward* dan *backpropagation*. *Feed-forward* merupakan tahap yang memproses data *input* melalui setiap *layer* yang ada hingga menjadi sebuah *output* yang diinginkan. *Backpropagation* adalah tahap kebalikan dari *feed-forward* yaitu proses menghitung dan menyesuaikan *weight* dan *bias* pada setiap *neuron* berdasarkan kesalahan pada hasil validasi *output*. Karena setiap *output* dari setiap *layer* akan saling berhubungan dengan *input layer* lain, maka hasil dari proses *backpropagation* setiap *neuron* di *output layer* akan merepresentasikan gradien keseluruhan yang dapat digabungkan dari seluruh *layer*. Melalui kedua proses tersebut model akan menyesuaikan parameter-parameter internal untuk menghasilkan *output* yang lebih akurat.

Proses *training* dan akurasi prediksi model *neural network* dipengaruhi oleh *hyperparameter*. *Hyperparameter* adalah parameter yang digunakan untuk mengontrol proses *training* pada model *machine learning*. Konfigurasi *hyperparameter* akan sangat berpengaruh pada proses *training* dan hasil akurasi model *machine learning* (Pannakkong dkk., 2022; Raji dkk., 2022). Secara umum *hyperparameter* pada *neural network* terdiri dari jumlah *input layer*, jumlah *hidden layer*, *learning rate*, dan *training cycles* (Pannakkong dkk., 2022).

a. Jumlah *neuron* pada *input layer*

Jumlah *neuron* pada *input layer* merupakan salah satu parameter yang mempengaruhi proses *training* dan akurasi prediksi. Hal ini karena semakin banyak *neuron* pada *input layer* berarti semakin banyak pula *neuron* yang harus dihubungkan dengan *neuron* di *hidden layer* berikutnya. Sebaliknya, semakin sedikit *neuron* pada *input layer* maka semakin sedikit pula *neuron* yang harus dihubungkan (Pannakkong dkk., 2022).

b. Jumlah *hidden layer*

Jumlah *hidden layer* yang ada pada sebuah *neural network* akan mempengaruhi proses *training* dan akurasi prediksi. Model yang memiliki *hidden layer* lebih kompleks mampu menerima dan mengolah relasi yang lebih kompleks antara *input* dan *output*. Namun hal ini juga memerlukan waktu yang lebih lama pada saat proses *training* model (Pannakkong dkk., 2022).

c. *Learning rate*

Learning rate adalah *hyperparameter* yang menentukan seberapa besar langkah yang diambil pada setiap iterasi saat proses *training* model. Pada *neural network* semakin kompleks relasi antara *input* dan *output layer* maka memerlukan *learning rate* yang semakin kecil. *Learning* rate yang semakin kecil memerlukan waktu *training* yang semakin lama, namun *learning rate* yang terlalu besar dapat menyebabkan model menjadi tidak akurat (Pannakkong dkk., 2022).

d. *Optimizer*

Optimizer merupakan algoritma yang digunakan untuk mengoptimalkan parameter model untuk mengurangi nilai dari *loss function* selama proses *training.* Salah satu *optimizer* yang paling sering digunakan dalam *neural network* saat ini adalah Adam. *Optimizer* Adam menggabungkan konsep dari momentum dan *adaptive learning rate* dengan menghitung rata-rata penurunan gradien secara eksponensial

untuk menyesuaikan *learning rate* pada setiap parameter. Adam menunjukkan performa yang baik pada berbagai permasalahan dan hingga saat ini menjadi salah satu algoritma pengoptimalan paling populer (Qi dkk., 2023).

e. *Batch size*

Batch size merupakan istilah yang mengacu pada jumlah contoh pelatihan data latih yang digunakan dalam satu iterasi. Hal yang perlu diperhatikan dalam menentukan nilai parameter adalah beban komputasi yang lebih besar apabila menggunakan nilai *batch size* yang semakin besar. Hal ini dikarenakan semakin besar nilai *batch size* maka semakin banyak pula *dataset* yang digunakan pada satu iterasi. Parameter ini merupakan salah satu *hyperparameter* terpenting untuk disesuaikan pada sistem *deep learning* (Rochmawati dkk., 2021).

f. *Training cycles*

Training cycles adalah jumlah iterasi dalam proses *training* model dan biasa disebut *epoch*. Dalam satu kali iterasi dilakukan penyesuaian nilai *weights* dan *biases* berdasarkan rekomendasi dari algoritma penurunan gradien. Durasi proses *training* akan berbanding lurus dengan jumlah iterasi yang dilakukan (Pannakkong dkk., 2022).

2.9. *Web Service*

Web service merupakan teknologi yang dapat menerima permintaan dan mengirim respon dari aplikasi lain dalam format XML atau JSON melalui internet. Umumnya *web service* memiliki ciri khusus berupa URL layaknya web, namun isi dari URL tersebut hanya mengandung sekumpulan informasi, perintah, dan konfigurasi tertentu (Wagito, 2022). *Web service* dapat terhubung dengan berbagai jenis aplikasi tanpa perlu memperhatikan *platform*, *database*, maupun bahasa pemrograman yang digunakan. *Web service* dapat bersifat privat maupun publik. *Web service* yang bersifat publik dapat diakses untuk layanan keperluan publik secara langsung,

namun *web service* yang bersifat privat hanya bisa diakses menggunakan *token* atau kunci tertentu.

Terdapat banyak *library* untuk membuat sebuah *web service*, salah satunya adalah FastAPI. FastAPI merupakan salah satu *library* bahasa pemrograman Python yang digunakan membuat *Application Programming Interface* (API). API adalah antarmuka yang digunakan untuk komunikasi dari satu aplikasi ke aplikasi lain (Hasanuddin dkk., 2022). Kelebihan dari FastAPI sendiri adalah dapat berjalan secara *asynchronous*, artinya FastAPI dapat berjalan tanpa perlu menunggu menerima permintaan. Selain itu FastAPI juga mudah diintegrasikan dengan Swagger dan ReDoc, yaitu *library* yang biasa digunakan untuk skema *OpenAPI* (Kornienko dkk., 2021).

2.10. *Confusion Matrix*

Confusion matrix adalah salah satu metode yang digunakan untuk mengevaluasi akurasi dari sebuah model klasifikasi. *Confusion matrix* berbentuk tabel yang digunakan untuk mengevaluasi kinerja dari suatu model klasifikasi. Tabel ini berisi informasi mengenai jumlah prediksi yang dilakukan menggunakan sebuah model. Kategori hasil prediksi *confusion* dalam bentuk tabel dapat dilihat pada Tabel 3.

Tabel 3. Kategori prediksi model pada *confusion matrix* (Bekkar dkk., 2013).

Prediksi / Sampel	Positif	Negatif
Positif	*True Positive*	*False Negative*
Negatif	*False Positive*	*True Negative*

Keterangan:

True Positive (TP) = Data positif yang terdeteksi positif

False Positive (FP) = Data positif yang terdeteksi negatif

True Negative (TN) = Data negatif yang terdeteksi negatif

False Negative (FN) = Data negatif yang terdeteksi positif

Dalam *confusion matrix* pengukuran performa secara umum dilakukan dengan menghitung nilai akurasi, presisi, *recall,* dan *F1-score* (Alpaydin, 2010).

a. Akurasi

Akurasi adalah nilai pengukuran paling umum untuk mengevaluasi efektivitas dari sebuah algoritma dengan mengestimasikan kemungkinan nilai benar dari sebuah kelas (Bekkar dkk., 2013). Akurasi menghitung seberapa akurat sebuah model dalam memprediksi data yang benar dengan cara membagi jumlah prediksi yang benar dengan jumlah data yang diprediksi. Persamaan akurasi dapat dilihat pada Persamaan 1.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (1)$$

b. Presisi

Presisi merupakan metode pengujian yang menghitung kemampuan model dalam mengidentifikasi data *false positive* (FP). Presisi dihitung dengan membagi jumlah prediksi *true positive* (TP) dengan jumlah prediksi *true positive* (TP) ditambah prediksi *false positive* (FP). Persamaan presisi dapat dilihat pada Persamaan 2.

$$precision = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

c. *Recall*

Recall merupakan metode pengujian yang menghitung akurasi model dalam memprediksi sampel yang bernilai benar. *Recall* dihitung dengan cara membagi jumlah prediksi *true positive* (TP) dengan jumlah sampel yang bernilai benar. Persamaan *recall* dapat dilihat pada Persamaan 3.

$$recall = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

d. *F1 Score*

F1 Score merupakan metode pengujian yang menghitung gabungan antara nilai presisi dan *recall*. *F1 Score* dihitung dengan cara membagi hasil perkalian nilai presisi dan *recall* dengan hasil penjumlahan nilai presisi dan *recall.* Persamaan *F1 score* dapat dilihat pada Persamaan 4.

$$F1\ Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (4)$$

III. METODOLOGI PENELITIAN

3.1. Waktu dan Tempat

a. Tempat Penelitian

Penelitian dilakukan di Laboratorium Komputasi Dasar Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

b. Waktu Penelitian

Penelitian ini dilaksanakan pada semester ganjil tahun akademik 2023/2024 dimulai dari bulan Oktober 2023 hingga bulan Desember 2023.

3.2. Perangkat Penelitian

a. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah sebuah laptop dengan dengan detail sebagai berikut.

Merk	:	Asus
Tipe	:	Vivobook 14X
Model	:	M1403QA
CPU	:	AMD Ryzen 5 5600H
GPU	:	AMD Radeon Vega 7 Graphics
RAM	:	16GB DDR4
Penyimpanan	:	SSD M.2 NVME Generasi 3 512GB
Kamera	:	USB2.0 HD UVC WebCam

b. Perangkat Lunak

Perangkat lunak yang digunakan dalam proses penelitian ini yaitu:

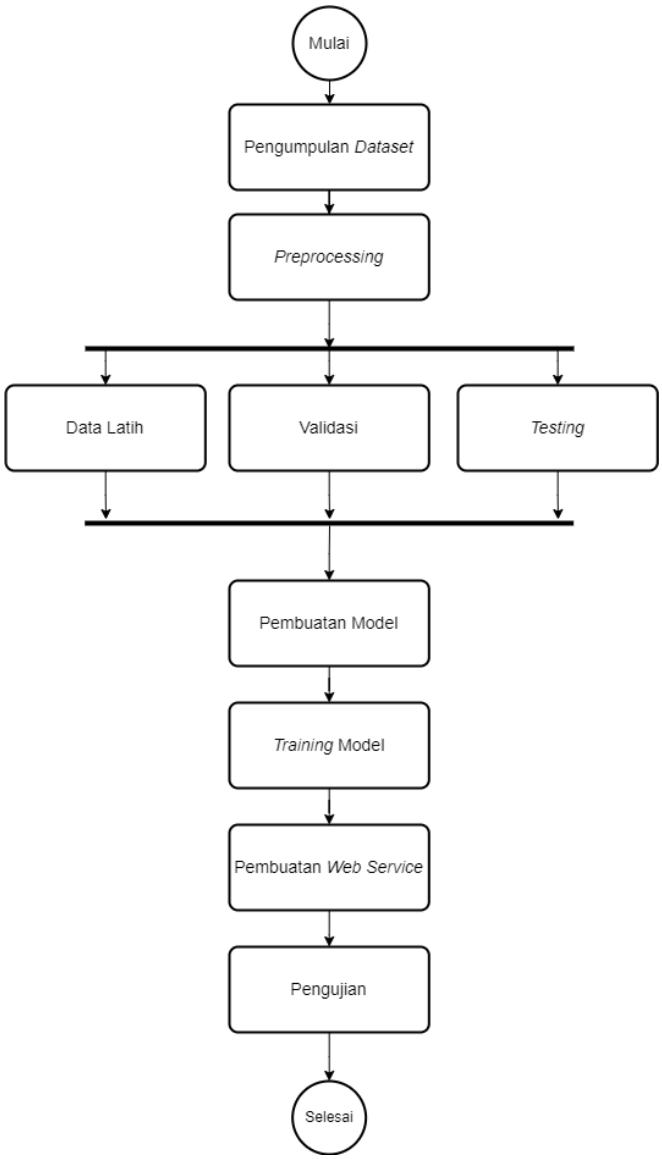
1. Sistem Operasi *Windows 11 Home 64-bit*
2. *Visual Studio Code* versi 1.83.1
3. *Python* versi 3.11.6 dengan *library Tensorflow* dan *FastAPI*
4. *Postman* versi 10.20

3.3. *Dataset*

Dataset yang digunakan dalam penelitian ini berupa kumpulan foto wajah mahasiswa Jurusan Ilmu Komputer Universitas Lampung. Data foto mahasiswa akan dipisahkan menjadi data latih, data validasi, dan data pengujian. Data ini dikumpulkan dari 20 mahasiswa yang masing-masing memiliki 20 foto wajah. Total *dataset* yang digunakan dalam penelitian ini adalah 400 foto. Selain itu, terdapat foto mahasiswa yang berbeda dari data latih. Data ini akan digunakan sebagai data pada pengujian *web service* untuk menguji penanganan terhadap foto mahasiswa yang tidak dikenali.

3.4. Tahap Penelitian

Tahapan yang akan dilakukan pada penelitian ini antara lain dimulai dari pengumpulan *dataset*, *preprocessing dataset*, pembagian *dataset* menjadi data *training* dan validasi, pembuatan model, *training* model, pembuatan *web service*, dan pengujian. Alur dari tahapan yang akan dilakukan pada penelitian ini diilustrasikan pada Gambar 10.



Gambar 10. Tahap penelitian.

a. Pengumpulan *Dataset*

Dataset pada penelitian ini diperoleh dari hasil wawancara yaitu meminta langsung kepada mahasiswa terkait. Metode ini dilakukan untuk mendapatkan data yang lebih beragam jika dibandingkan dengan mengambil gambar secara langsung ketika wawancara. Dari hasil pengumpulan tersebut diperoleh 20 foto dari 20 mahasiswa Jurusan Ilmu Komputer Universitas Lampung, serta foto mahasiswa yang berbeda dari 20 mahasiswa sebelumnya.

b. *Preprocessing Dataset*

Preprocessing perlu dilakukan untuk meningkatkan kualitas *dataset* agar meningkatkan efisiensi dan efektifitas pada saat *training* model dilakukan. *Preprocessing* yang dilakukan adalah mendeteksi dan memotong foto pada bagian wajah, lalu foto tersebut diubah ukurannya (*resize*) menjadi 224x224 *pixel*. Foto yang telah diubah ukurannya kemudian melalui proses augmentasi yaitu proses memperbanyak variasi *dataset* berdasarkan *dataset* yang sudah ada. Metode augmentasi yang digunakan adalah *grayscale, histogram equalization, logarithmic transform, gamma correction,* dan *random erasing*. Setiap satu foto yang akan digunakan sebagai *dataset training* dan *validation* akan diperbanyak menjadi 4 foto yang masing-masing adalah hasil dari proses *grayscale*, *histogram equalization, logarithmic transform,* dan *gamma correction*. Hasil dari proses tersebut kemudian akan diperbanyak lagi menggunakan metode *random erasing*. Setiap satu foto akan diperbanyak menjadi 5 foto yang terdiri dari 1 foto asli dan 4 foto yang telah melalui proses *random erasing*.

c. Pemisahan Data

Dataset yang telah melalui tahap *preprocessing* kemudian dibagi menjadi data uji, data latih, dan data validasi.

1) Data uji

Data uji atau data *testing* merupakan data yang digunakan dalam tahap pengujian pada model. Data uji yang digunakan pada penelitian ini terdiri dari dua jenis data, yaitu data uji yang harus dikenali dan data uji yang harus tidak dikenali oleh model. Data uji yang harus dikenali diambil sebanyak 10% dari total *dataset* utama atau sebanyak 2 foto dari masing-masing mahasiswa, dan dipisahkan terlebih dahulu dari *dataset* utama sebelum pemisahan data latih dan data validasi.

2) Data latih

Data latih adalah data yang digunakan pada tahap *training* model. Data latih yang digunakan pada penelitian ini diambil secara acak dari *dataset* utama. Pengambilan data latih dilakukan setelah data uji dipisahkan terlebih dahulu dari *dataset* utama. Pemisahan data dilakukan menggunakan fungsi *image_dataset_from_directory* dari *library tensorflow* sebanyak 80% dari total *dataset*.

3) Data validasi

Data validasi adalah data yang digunakan pada tahap *training* model untuk melakukan validasi terhadap hasil *training* pada setiap iterasi dan dilakukan untuk mencegah *overfitting*. Data uji yang digunakan pada penelitian ini diambil secara acak sebanyak 20% dari *dataset* utama. Pengambilan data validasi dilakukan setelah data uji dipisahkan terlebih dahulu dari *dataset* utama dan dilakukan menggunakan fungsi *image_dataset_from_directory* dari *library tensorflow*.

d. *Training Model*

Training model dalam penelitian ini dilakukan menggunakan dua model yaitu ResNet50 dan ResNet101 dengan data yang akan dilatih sebanyak 14,400 foto mahasiswa. *Training* dilakukan menggunakan model parameter yang sama pada kedua model. *Hyperparameter* yang digunakan pada proses *training* dapat dilihat pada Tabel 4. Model yang dihasilkan dari proses *training* akan disimpan dalam sebuah *file* dengan ekstensi h5 (.h5).

Tabel 4. *Hyperparameter training* model.

Nama Parameter	Nilai
Input size	224x224x3
Batch size	8
Number of classes	20
Optimizer	Adam

e. Pengembangan *Web Service*

Web Service yang akan dikembangkan akan menggunakan *library* FastAPI. *Library* ini dipilih karena lebih mudah dalam integrasi dengan modul eksternal. Modul eksternal yang akan digunakan adalah modul *tensorflow*. Modul ini digunakan untuk melakukan prediksi menggunakan modul *machine learning* yang sebelumnya telah disimpan dalam *file* dengan ekstensi h5 (.h5).

Untuk melakukan prediksi menggunakan model *machine learning*, *web service* perlu menerima sebuah *file* gambar yang dikirim melalui *endpoint* "(*base_url*)/api/v1/predict". Gambar yang diterima kemudian akan diubah ukurannya (*resize)* menjadi berukuran 224x224. Hal ini dilakukan untuk menyesuaikan ukuran gambar dengan ukuran *input*

yang digunakan model. Ketika gambar sudah disesuaikan, maka model akan melakukan prediksi terhadap gambar. Hasil prediksi dari model akan dikirim sebagai sebuah respon kepada klien dalam format JSON.

f. Pengujian

Pengujian dilakukan untuk melihat apakah model dan *Web Service* yang dibuat memenuhi kriteria yang diinginkan. Pengujian yang dilakukan terdiri dari dua jenis pengujian, yaitu pengujian model dan pengujian *web service*.

1) Pengujian model

Pengujian model dilakukan menggunakan *confusion matrix* untuk menghitung akurasi, presisi, *recall*, dan *specificity* dari model yang telah dibuat. Pengujian dilakukan menggunakan *dataset* pengujian yang sebelumnya telah dipisahkan dari *dataset training* dan validasi. Pengujian model akan dibedakan menjadi dua, yaitu pengujian akurasi model terhadap *dataset* yang sudah dilatih dan kemampuan model dalam menangani foto mahasiswa yang tidak termasuk dalam *dataset training*. Pengujian akurasi model akan dilakukan menggunakan *dataset testing* yang dikumpulkan dengan mengambil dua foto dari setiap mahasiswa yang dilatih pada model. Total *dataset testing* yang digunakan pada pengujian akurasi berjumlah 40 foto.

2) Pengujian *web service*

Pengujian *web service* dilakukan untuk memastikan bahwa *web service* yang dibuat telah memenuhi kebutuhan yang diperlukan. Pengujian *web* service dilakukan dengan mengirimkan permintaan ke *endpoint* yang telah dibuat menggunakan aplikasi *Postman.* Hasil yang diperoleh berupa respon dari *web service* akan dicatat

dan dibandingkan dengan respon yang diharapkan. Pengujian akan dilakukan dengan beberapa skenario yang telah ditentukan. Skenario pengujian dan hasil yang diharapkan tertera pada tabel 6.

Tabel 5. Skenario pengujian *web service.*

Endpoint	**Deskripsi**	***Input***	**Respon yang diharapkan**
/api/v1/predict	Mengirimkan sebuah foto wajah mahasiswa yang telah dilatih pada model *machine learning.*	*File* foto dengan ekstensi .jpg atau .png.	Nomor Pokok Mahasiswa (NPM) dari mahasiswa yang fotonya dikirimkan sebagai *input*.
/api/v1/predict	Mengirimkan foto wajah mahasiswa yang belum dilatih pada model *machine learning*.	*File* foto dengan ekstensi .jpg atau .png	Foto tidak dikenali.
/api/v1/predict	Mengirimkan sebuah *file* dengan ekstensi bukan gambar.	*File* dengan ekstensi .pdf.	*Error* ekstensi *file* tidak sesuai (415).
/api/v1/prediksi	Mengirimkan foto mahasiswa pada *endpoint* yang tidak ada pada *web service.*	*File* foto dengan ekstensi .jpg atau .png.	*Error endpoint* tidak ditemukan (404).

V. SIMPULAN DAN SARAN

5.1. Simpulan

Simpulan dari penelitian yang telah dilakukan sebagai berikut:

1. Penelitian ini berhasil membuat model CNN dengan arsitektur ResNet50 dan ResNet101, namun model yang dibuat hanya mendapatkan akurasi sebesar 60% pada saat pengujian.
2. Nilai akurasi model yang kurang baik pada pengujian diduga karena arsitektur ResNet50 dan ResNet101 memerlukan jumlah *dataset training* yang banyak untuk mendapatkan nilai akurasi yang tinggi.
3. Penelitian ini juga telah berhasil membuat *web service* yang dapat digunakan menjadi perantara model dengan aplikasi *client* untuk melakukan prediksi pengenalan wajah untuk sistem presensi.

5.2. Saran

Saran yang dapat diberikan untuk melanjutkan penelitian ini sebagai berikut:

1. Melanjutkan pengembangan model CNN dengan arsitektur ResNet50 dan ResNet101 dengan *dataset* yang lebih banyak.
2. Mengembangkan sistem presensi yang dapat menjadi antarmuka pengguna untuk melakukan pengenalan wajah.
3. Mengembangkan kamera berbasis *Internet of Things* yang dapat mendukung sistem presensi untuk melakukan pencatatan kehadiran secara otomatis.

DAFTAR PUSTAKA

Albawi, Mohammed, & AL-Zawi. 2017. Understanding of a Convolutional Neural Network. In: *International Conference on Engineering and Technology (ICET)*. IEEE pp. 1–6.

Alpaydin. 2010. *Introduction to Machine Learning*. The MIT Press, Cambridge. 579 hlm.

Bekkar, Djemaa, & Alitouche. 2013. Evaluation Measures for Models Assessment over Imbalanced Data Sets. *J Inf Eng Appl*. 3 : 1–13.

Carleo, Cirac, Cranmer, Daudet, Schuld, Tishby, Vogt-Maranto, & Zdeborová. 2019. Machine Learning and the Physical Sciences. *Reviews of Modern Physics*. 91 : 045002.

Detila, Eri, & Wibowo. 2019. Perbandingan Metode Eigenface, Fisherface, dan LBPH pada Sistem Pengenalan Wajah. *Jurnal Ilmiah Komputasi*. 18 : 315–322.

Elpeltagy & Sallam. 2021. Automatic Prediction of COVID−19 from Chest Images Using Modified ResNet50. *Multimedia Tools and Applications*. 80 : 26451–26463.

Haryanto, Rahaningsih, & Basysyar. 2023. Komparasi Algoritma Machine Learning Dalam Memprediksi Harga Rumah. *Jurnal Mahasiswa Teknik Informatika*. 7 : 533.

Hasanuddin, Asgar, & Hartono. 2022. Rancang Bangun REST API Aplikasi WeShare Sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan. *JINTEKS (Jurnal Informatika Teknologi dan Sains)*. 4 : 8–14.

He, Zhang, Ren, & Sun. 2016. Deep Residual Learning for Image Recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* pp. 770–778.

Kornienko, Mishina, Shcherbatykh, & Melnikov. 2021. Principles of Securing RESTful API Web Services Developed with Python Frameworks. *Journal of Physics: Conference Series*. 2094

Mandal, Okeukwu, & Theis. 2021. Masked Face Recognition using ResNet-50. *arXiv preprint arXiv*. 2104–08997.

Masruroh, Surarso, Warsito, & Korespondensi. 2023. Perbandingan Kinerja Inception-ResNetV2, XCeption, Inception-V3, dan ResNet50 pada Gambar Bentuk Wajah. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*. 10 : 11–20.

Niswati, Hardatin, Muslimah, & Hasanah. 2021. Perbandingan Arsitektur ResNet50 dan ResNet101 dalam Klasifikasi Kanker Serviks pada Citra Pap Smear. *Faktor Exacta*. 14 : 160.

Nugroho, Fenriana, & Arijanto. 2020. Implementasi Deep Learning Menggunakan Convolutional Neural Network pada Ekspresi Manusia. *Jurnal Algor*. 2 : 12–20.

Oktaviana, Hendrawan, Annas, & Wicaksono. 2021. Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*. 5 : 1216–1222.

Pannakkong, Thiwa-Anont, Singthong, Parthanadee, & Buddhakulsomsiri. 2022. Hyperparameter Tuning of Machine Learning Algorithms Using Response Surface Methodology: A Case Study of ANN, SVM, and DBN. *Mathematical Problems in Engineering*. 2022 : 1–17.

Qi, Wang, & Zhang. 2023. Understanding Optimization of Deep Learning via Jacobian Matrix and Lipschitz Constant. *arXiv preprint arXiv*. 2306–09338.

Raji, Bello-Salau, Umoh, Onumanyi, Adegboye, & Salawudeen. 2022. Simple Deterministic Selection-Based Genetic Algorithm for Hyperparameter Tuning of Machine Learning Models. *Applied Sciences (Switzerland)*. 12 : 1186.

Ramdhon & Febriya. 2021. Penerapan Face Recognition Pada Sistem Presensi. *Journal of Applied Computer Science and Technology*. 2 : 12–17.

Rochmawati, Hidayati, Yamasari, Peni, Tjahyaningtijas, Yustanti, & Prihanto. 2021. Analisa Learning rate dan Batch size Pada Klasifikasi Covid Menggunakan Deep learning dengan Optimizer Adam. *Journal Information Engineering and Educational Technology*. 05 : 44–48.

Samuel. 1959. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*. 3 : 210–229.

Setyawan, Nilogiri, & A'yun. 2023. Implementasi Convolution Neural Network (CNN) untuk Klasifikasi pada Citra Ikan Cupang Hias. *Jurnal Teknik Informatika Kaputama (JTIK)*. 7 : 101–110.

Song, Yang, & Zhang. 2021. Does Preprocessing Help Training Over-parameterized Neural Networks? *Advances in Neural Information Processing Systems*. 34 : 22890–22904.

Wagito. 2022. Implementasi Web Service Waktu Shalat Berbasis Teknologi PAAS Cloud Computing. *Technologia*. 13 : 331–338.

Wajdi & Sugiantara. 2018. Pemanfaatan Teknik Pengenalan Wajah Berbasis Opencv untuk Sistem Informasi Pencatatan Kehadiran Dosen. *Infotek : Jurnal Informatika dan Teknologi*. 1 : 96–106.

Wijaya, Samodra, & Suyoto. 2023. Sistem Presensi Pegawai dengan Face Recognition Menggunakan Deep Learning CNN. *Jurnal Informatika Atma Jogja*. 4 : 163–168.