

**KLASIFIKASI CITRA SAMPAH PADAT MENGGUNAKAN
MODEL *DEEP RESIDUAL NETWORK* (RESNET-50)**

(Skripsi)

Oleh

ALFA RIZKI FADLILLAH

NPM 2117051003



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

KLASIFIKASI CITRA SAMPAH PADAT MENGGUNAKAN MODEL *DEEP RESIDUAL NETWORK (RESNET-50)*

Oleh

ALFA RIZKI FADLILLAH

Pengelolaan sampah padat menjadi tantangan global yang mendesak, diperparah oleh keterbatasan proses pemilahan manual yang lambat, tidak efisien, dan berisiko bagi kesehatan. Otomatisasi pemilahan sampah melalui teknologi *deep learning* menawarkan solusi yang menjanjikan. Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi model *deep learning* menggunakan arsitektur *Deep Residual Network* (ResNet-50) untuk klasifikasi citra sampah padat secara otomatis. Metodologi penelitian ini menggunakan dataset publik dari Nnamoko *et al.* (2022) yang terdiri dari 24.705 citra yang dikategorikan ke dalam dua kelas: *organic* dan *recyclable*. Tahapan penelitian meliputi *preprocessing* data dengan mengubah ukuran citra menjadi 224×224 piksel dan normalisasi, diikuti dengan augmentasi data ekstensif menggunakan ImageDataGenerator untuk meningkatkan jumlah data latih menjadi 207.522 citra guna mencegah *overfitting*. Model ResNet-50 yang telah dilatih sebelumnya (*pre-trained*) pada dataset *ImageNet* diadaptasi menggunakan teknik *transfer learning* dan dilatih ulang pada dataset sampah. Hasil evaluasi pada data uji menunjukkan bahwa model yang dikembangkan berhasil mencapai tingkat akurasi sebesar 95%. Metrik kinerja lainnya seperti presisi, sensitivitas (*recall*), dan F1-score juga menunjukkan hasil yang sangat baik, dengan rata-rata tertimbang 0.95. Hasil ini membuktikan bahwa model ResNet-50 memiliki efektivitas tinggi untuk tugas klasifikasi citra sampah padat dan menawarkan solusi yang lebih akurat dibandingkan model *Convolutional Neural Network* (CNN) yang lebih sederhana, sehingga berpotensi besar untuk diimplementasikan dalam sistem pemilahan sampah otomatis.

Kata Kunci: *Deep Learning*, Klasifikasi Citra, Klasifikasi Sampah, *Residual Network*, ResNet-50, Sampah Padat, *Transfer Learning*.

ABSTRACT

CLASSIFICATION OF SOLID WASTE IMAGES USING DEEP RESIDUAL NETWORK MODEL (RESNET-50)

By

ALFA RIZKI FADLILLAH

Solid waste management is an urgent global challenge, exacerbated by the limitations of manual sorting processes, which are slow, inefficient, and pose health risks. Automating waste sorting through deep learning technology offers a promising solution. This study aims to develop and evaluate a deep learning model using the Deep Residual Network (ResNet-50) architecture for automatic classification of solid waste images. The research methodology utilizes a public dataset from Nnamoko et al. (2022), consisting of 24,705 images categorized into two classes: organic and recyclable. The research stages include data preprocessing by resizing the images to 224×224 pixels and normalization, followed by extensive data augmentation using ImageDataGenerator to increase the training data to 207,522 images to prevent overfitting. The pre-trained ResNet-50 model on the ImageNet dataset was adapted using transfer learning techniques and retrained on the waste dataset. Evaluation results on the test data showed that the developed model achieved an accuracy rate of 95%. Other performance metrics such as precision, sensitivity (recall), and F1-score also showed excellent results, with a weighted average of 0.95. These results demonstrate that the ResNet-50 model is highly effective for solid waste image classification tasks and offers a more accurate solution compared to simpler Convolutional Neural Network (CNN) models, making it highly promising for implementation in automated waste sorting systems.

Keywords: Deep Learning, Image Classification, Residual Network, ResNet-50, Solid Waste, Transfer Learning, Waste Classification.

**KLASIFIKASI CITRA SAMPAH PADAT MENGGUNAKAN
MODEL *DEEP RESIDUAL NETWORK* (RESNET-50)**

Oleh

ALFA RIZKI FADLILLAH

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA ILMU KOMPUTER**

Pada

**Program Studi Ilmu Komputer
Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

Judul Skripsi : **KLASIFIKASI CITRA SAMPAH PADAT
MENGGUNAKAN MODEL DEEP
RESIDUAL NETWORK (RESNET-50)**

Nama Mahasiswa : **Alfa Rizki Fadlillah**

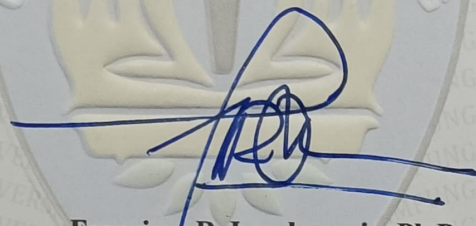
Nomor Pokok Mahasiswa : 2117051003

Program Studi : S1 Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

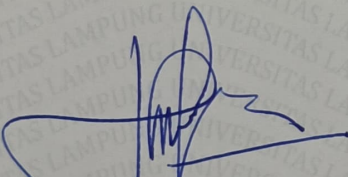


Favorisen R. Lumbanraja, Ph.D.

NIP. 19830110 200812 1 002

2. Mengetahui

Ketua Jurusan Ilmu Komputer



Dwi Sakethi, S.Si., M.Kom.

NIP. 19680611 199802 1 001

Ketua Program Studi

S1 Ilmu Komputer



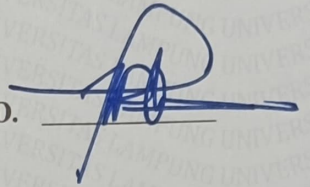
Tristiyanto, M.I.S., Ph.D.

NIP. 19810414 200501 1 001

MENGESAHKAN

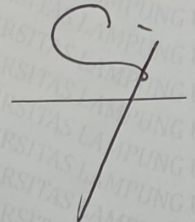
1. Tim Penguji

Ketua : Favorisen R. Lumbanraja, Ph.D.



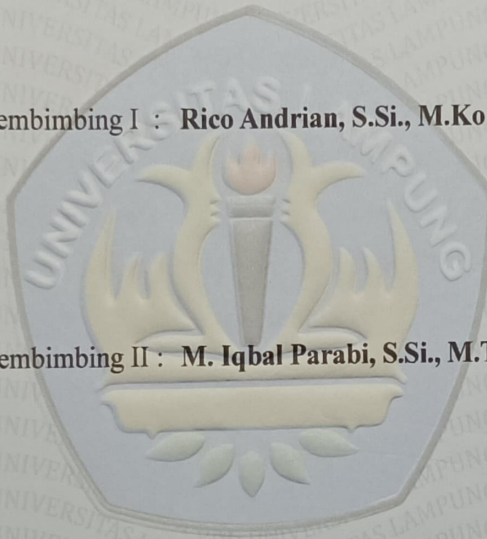
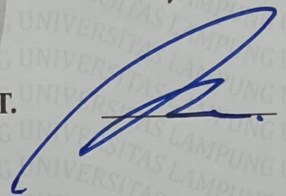
Penguji

Bukan Pembimbing I : Rico Andrian, S.Si., M.Kom.

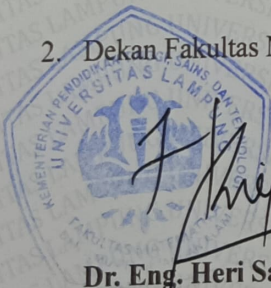


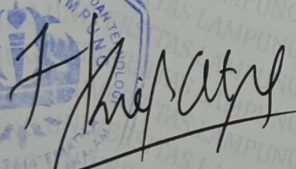
Penguji

Bukan Pembimbing II : M. Iqbal Parabi, 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: 19 Juni 2025

PERNYATAAN

Saya bertanda tangan di bawah ini:

Nama : Alfa Rizki Fadlillah

NPM : 2117051003

Dengan ini menyatakan bahwa skripsi saya yang berjudul "**Klasifikasi Citra Sampah Padat Menggunakan Model *Deep Residual Network* (ResNet-50)**" merupakan hasil karya saya sendiri dan bukan karya orang lain. Seluruh 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, 11 Juli 2025



Alfa Rizki Fadlillah
NPM. 2117051003

RIWAYAT HIDUP



Penulis dilahirkan pada 29 Oktober 2002 di Kota Bandar Lampung, Provinsi Lampung sebagai anak kedua dari pasangan Bapak Aliman M. Rasyid dan Ibu Farida Irayani.

Penulis Pendidikan di Taman Kanak-Kanak (TK) Sriwijaya pada tahun 2008 dan melanjutkan pendidikan di Madrasah Ibtidaiyah Negeri 5 Bandar Lampung pada tahun 2009-2015. Kemudian penulis melanjutkan pendidikan menengah pertama di Madrasah Tsanawiyah Negeri 2 Bandar Lampung pada tahun 2015-2018, dan pendidikan menengah atas di Madrasah Aliyah Negeri 1 Bandar Lampung di tahun 2018-2021.

Pada Tahun 2021 penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan antara lain:

1. Asisten Dosen Jurusan Ilmu Komputer pada Tahun Ajaran 2023/2024.
2. Pada Bulan Januari 2024 melaksanakan KKN di Desa Cilimus, Kecamatan Teluk Pandan, Kabupaten Pesawaran.
3. Pada bulan Agustus 2024 penulis melaksanakan Kerja Praktik di PT. PLN NP UP Bandar Lampung.

PERSEMBAHAN

Alhamdulillahirabbil'alamiin

Puji syukur kehadiran Allah Subhanahu Wa Ta'ala atas segala rahmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan dengan sebaik-baiknya. Shalawat serta salam senantiasa tercurahkan kepada suri teladan Nabi Muhammad Shallallahu 'Alaihi Wasallam.

Kupersembahkan karya ini kepada:

Ayah, Ibuku, Kakak dan Adik Tercinta

Atas setiap doa, dukungan, dan kasih sayang yang tak pernah putus.
Terima kasih atas pengorbanan, semangat, dan kepercayaan yang telah menjadi kekuatan terbesar selama menempuh pendidikan ini.

Seluruh Keluarga Besar Ilmu Komputer 2021

Yang telah memberikan ilmu, bantuan, dan dukungan selama masa perkuliahan.

Almamater Tercinta, Universitas Lampung dan Jurusan Ilmu Komputer

Tempat bertumbuh, belajar, dan mendapatkan bekal hidup.

SANWACANA

Segala puji dan syukur penulis panjatkan ke hadirat Allah Subhanahu Wa Ta'ala atas segala limpahan nikmat, rahmat, dan hidayah-Nya sehingga penulisan skripsi ini dapat terselesaikan dengan baik dan lancar. Shalawat serta salam semoga senantiasa tercurah kepada Nabi besar Muhammad Shallallahu 'Alaihi Wasallam, pembawa risalah kebenaran bagi umat manusia. Skripsi berjudul "**Klasifikasi Citra Sampah Padat Menggunakan Model *Deep Residual Network (ResNet-50)***" telah disusun dengan sebaik-baiknya dan sebagai salah satu syarat untuk mendapatkan gelar sarjana ilmu komputer di Universitas Lampung.

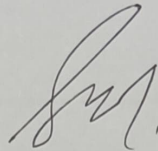
Penulis menyadari bahwa dalam proses penyusunan skripsi ini tidak terlepas dari dukungan, bimbingan, serta bantuan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah Subhanahu Wa Ta'ala, yang senantiasa melimpahkan nikmat, petunjuk, serta ridho-Nya, sehingga penulisan karya ilmiah ini dapat diselesaikan dengan baik dan tepat waktu.
2. Ayah, Ibu, Kakak dan Adik yang selalu memberikan cinta, doa, semangat, serta dukungan tanpa henti. Terima kasih atas segala pengorbanan, motivasi, dan perhatian yang tulus, yang menjadi sumber kekuatan penulis dalam menyelesaikan pendidikan dan penulisan skripsi ini.
3. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan FMIPA Universitas Lampung.
4. Bapak Dwi Sakethi, S.Si., M. Kom. selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.

5. Ibu Yunda Heningtyas, M. Kom. selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
6. Bapak Tristiyanto, S. Kom., M.I.S., Ph.D. selaku Ketua Program Studi Ilmu Komputer FMIPA Universitas Lampung.
7. Bapak Prof. Admi Syarif, Ph.D. selaku Pembimbing Akademik penulis, yang senantiasa memberikan dukungan dan arahan dalam pengembangan akademik selama masa studi.
8. Bapak Favorisen Rosyking Lumbanraja, Ph.D. selaku dosen Pembimbing Utama, yang telah dengan sabar dan penuh dedikasi membimbing penulis selama proses penyusunan skripsi ini. Terima kasih atas waktu, ilmu, arahan, serta koreksi yang sangat berarti dalam menyempurnakan karya ini.
9. Bapak Rico Andrian, S.Si., M.Kom. selaku dosen Pembahas, yang telah memberikan arahan, masukan, dan motivasi yang sangat membantu dalam proses penyusunan skripsi ini. Terima kasih atas kesediaan, ketelitian, serta perhatian Bapak dalam membimbing penulis hingga skripsi ini dapat terselesaikan dengan baik.
10. Bapak M. Iqbal Parabi, S.Si., M.T. selaku dosen Pembahas, yang telah memberikan kritik, saran, dan masukan konstruktif yang sangat berarti bagi penyempurnaan skripsi ini. Terima kasih atas perhatian dan kontribusi Ibu dalam proses akademik penulis.
11. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung, yang telah membagikan ilmu, motivasi, serta pengalaman berharga selama penulis menempuh pendidikan di Jurusan Ilmu Komputer Universitas Lampung.
12. Seluruh staf dan karyawan Jurusan Ilmu Komputer, Ibu Ade Nora Maela, Bang Zainuddin, Mas Syam, dan Mas Nofal, yang telah membantu kelancaran berbagai urusan administrasi, laboratorium, serta peminjaman ruangan selama masa perkuliahan penulis di Jurusan Ilmu Komputer.

Penulis menyadari bahwa skripsi ini masih memiliki berbagai kekurangan dan belum mencapai kesempurnaan. Namun demikian, penulis berharap karya ini dapat memberikan manfaat, khususnya bagi civitas akademika Universitas Lampung, serta menjadi referensi dan kontribusi positif bagi mahasiswa Ilmu Komputer.

Bandar Lampung, 11 Juli 2025



Alfa Rizki Fadlillah
NPM. 2117051003

DAFTAR ISI

	Halaman
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
DAFTAR KODE PROGRAM	x
I. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
II. TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu	6
2.2 Sampah	9
2.2.1 Sampah Padat.....	10
2.3 Klasifikasi Gambar	10
2.4 <i>Machine Learning</i>	11
2.5 <i>Deep Learning</i>	11
2.5.1 <i>Input Layer</i>	12
2.5.2 <i>Hidden Layer</i>	12
2.5.3 <i>Output Layer</i>	13
2.6 <i>Convolutional Neural Network (CNN)</i>	13
2.6.1 <i>Convolutional Layer</i>	14
2.6.2 <i>Activation Function</i>	14
2.6.3 <i>Pooling Layer</i>	17

2.6.4	<i>Fully Connected Layer</i>	17
2.6.5	<i>Dropout Layer</i>	18
2.7	<i>Vanishing Gradient</i>	19
2.8	<i>Residual Network (ResNet)</i>	19
2.9	Ekstraksi Fitur	21
2.10	<i>Dataset Splitting</i>	22
2.11	Augmentasi Data	23
2.11.1	<i>ImageDataGenerator</i>	24
2.12	<i>Confusion Matrix</i>	24
2.12.1	Akurasi	25
2.12.2	Presisi	26
2.12.3	F1-Score	26
2.12.4	Sensitivitas	26
2.12.5	Spesifisitas	27
2.13	<i>Underfitting</i>	27
2.14	<i>Overfitting</i>	28
III.	METODOLOGI PENELITIAN	29
3.1	Tempat dan Waktu Penelitian	29
3.1.1	Tempat Penelitian	29
3.1.2	Waktu Penelitian	29
3.2	Data dan Alat	31
3.2.1	Data	31
3.2.2	Alat	32
3.3	Tahap Penelitian	34
3.3.1	Pengumpulan Data	35
3.3.2	<i>Pre-Processing</i>	35
3.3.3	Pembagian Data	35
3.3.4	Augmentasi Data	36
3.3.5	Pembuatan Model	36
3.3.6	Pengujian dan Evaluasi	37
IV.	HASIL DAN PEMBAHASAN	38
4.1	Pengumpulan Dataset	38
4.2	<i>Pre-Processing</i>	38

4.3	Pembagian Data.....	39
4.4	Augmentasi Data	39
4.5	Pembuatan Model.....	43
4.5.1	Arsitektur Model	43
4.5.2	<i>Training</i> Model	45
4.6	Evaluasi	46
4.6.1	Hasil Klasifikasi	46
4.6.2	Visualisasi Pelatihan.....	48
4.6.3	<i>Confusion Matrix</i>	49
V.	SIMPULAN DAN SARAN	52
5.1	Simpulan.....	52
5.2	Saran	53
	DAFTAR PUSTAKA.....	54

DAFTAR GAMBAR

Gambar	Halaman
1. Contoh sampah (Borut, 2022).....	9
2. <i>Neural Network Layer</i> (GeeksforGeeks, 2025).	12
3. Struktur CNN (Dharmaraj, 2022).	13
4. <i>Convolutional Layer</i> (Bhatt <i>et al.</i> , 2021).	14
5. Ilustrasi <i>ReLU</i> (Wang <i>et al.</i> , 2020).....	15
6. Ilustrasi <i>Sigmoid</i> (Wang <i>et al.</i> , 2020).....	16
7. Ilustrasi <i>Tanh</i> (Wang <i>et al.</i> , 2020).	16
8. <i>Max Pooling</i> (Bhatt <i>et al.</i> , 2021).....	17
9. <i>Average Pooling</i> (Bhatt <i>et al.</i> , 2021).....	17
10. <i>Fully Connected Layer</i> (Bhatt <i>et al.</i> , 2021).	18
11. Ilustrasi <i>underfitting</i> (IBM, 2024).....	27
12. Ilustrasi <i>overfitting</i> (IBM, 2024).....	28
13. Contoh data <i>organic</i> (a) dan contoh data <i>recyclable</i> (b).	31
14. Rasio data <i>organic</i> dan <i>recyclable</i>	31
15. Tahapan Penelitian (Nnamoko <i>et. al.</i> , 2022).	34
16. Contoh gambar dari masing masing kelas.	38
17. Contoh gambar dan hasil augmentasi.	42
18. Arsitektur ResNet-50 (Nitish, 2023).	44
19. Akurasi Pelatihan dan Validasi.	48
20. <i>Loss</i> Pelatihan dan Validasi.	49
21. <i>Confusion Matrix</i> Hasil Klasifikasi.....	50

DAFTAR TABEL

Tabel	Halaman
1. Penelitian terdahulu terkait klasifikasi sampah.....	6
2. Arsitektur ResNet50 (He <i>et al.</i> , 2016).....	20
3. <i>Binary Confusion Matrix</i>	25
4. Alur pelaksanaan penelitian	30
5. Parameter augmentasi.	36
6. <i>Hyperparameter training</i>	37
7. Distribusi dataset per kelas.....	39
8. Distribusi data latih setelah proses augmentasi.....	40
9. Parameter augmentasi dan deskripsinya.	40
10. <i>Hyperparameter</i> yang sudah di- <i>tuning</i>	45
11. <i>Classification Report</i>	47

DAFTAR KODE PROGRAM

Kode Program	Halaman
1. Kode implementasi augmentasi data latih.....	42
2. Kode implementasi model.....	44
3. Kode implementasi <i>training</i> model.....	46

I. PENDAHULUAN

1.1 Latar Belakang

Sampah padat merupakan permasalahan global yang terus berkembang seiring berjalannya waktu. Berdasarkan data dari *World Bank*, pada tahun 2020 produksi sampah global diperkirakan mencapai 2,01 miliar ton per tahun dan diproyeksikan akan meningkat hingga 3,4 miliar ton pada tahun 2050, seiring dengan peningkatan populasi dan urbanisasi (Kaza *et al.*, 2018). Sampah padat yang tidak dikelola dengan baik dapat menyebabkan pencemaran lingkungan termasuk tanah, udara, dan air, serta berdampak pada lingkungan disekitarnya. Untuk itu, diperlukan pengembangan alat yang dapat meningkatkan otomatisasi pengelolaan sampah (Nnamoko *et al.*, 2022), sehingga dapat meningkatkan efisiensi dalam pemilahan dan daur ulang sampah padat.

Proses pemilahan sampah secara manual masih menjadi praktik utama di banyak negara, terutama di negara berkembang. Namun, pendekatan ini memiliki berbagai kendala signifikan, seperti pemilahan manual sangat bergantung pada tenaga manusia yang tidak hanya menurunkan efisiensi tetapi juga menimbulkan risiko kesehatan seperti gangguan muskuloskeletal akibat postur kerja yang buruk (Emmatty *and* Panicker, 2022). Selanjutnya, proses ini juga cenderung lambat dan tidak mampu mengikuti laju peningkatan volume dan kompleksitas sampah saat ini, sehingga berdampak pada rendahnya tingkat daur ulang (Satav *et al.*, 2023). Efektivitas pemilahan manual sangat dipengaruhi oleh faktor manusia, seperti kelelahan dan kesalahan dalam pengenalan jenis sampah, yang dapat menyebabkan pencampuran material daur ulang dan non-daur ulang (Andrade *and* Mondelli,

2024). Oleh karena itu, mempercepat transisi ke sistem pemilahan yang lebih otomatis dan cerdas menjadi semakin mendesak guna mencapai efisiensi dan keberlanjutan dalam pengelolaan sampah padat.

Perkembangan teknologi kecerdasan buatan (*Artificial Intelligence*) telah membawa perubahan di berbagai bidang, termasuk dalam pengelolaan sampah padat. Kecerdasan buatan khususnya *deep learning*, menawarkan solusi yang lebih efisien dalam analisis citra digital dan pengenalan pola yang kompleks. *Deep learning* yang merupakan cabang dari *machine learning* mampu mengolah data dalam jumlah besar dan mempelajari pola data tersebut tanpa campur tangan manusia. Penerapannya dalam klasifikasi citra telah menghasilkan model yang sangat akurat dalam mengenali objek-objek spesifik, salah satunya adalah dalam mengklasifikasikan berbagai jenis sampah (Lecun *et al.*, 2015). Kemampuan kecerdasan buatan untuk memproses citra digital secara cepat dan akurat membuka peluang besar dalam memecahkan masalah pengelolaan sampah secara otomatis.

Deep Residual Network (ResNet) merupakan salah satu arsitektur jaringan saraf tiruan yang populer dalam klasifikasi citra. Diperkenalkan oleh He *et al.* (2016), ResNet-50 adalah versi dengan 50 lapisan yang menggunakan pendekatan *residual learning* untuk mengatasi masalah *vanishing gradient* pada pelatihan jaringan *neural* yang sangat dalam. Keunggulan utama dari ResNet-50 adalah kemampuannya untuk menjaga akurasi meski menggunakan jaringan yang sangat dalam, menjadikannya model yang ideal untuk klasifikasi gambar dengan berbagai tingkat kompleksitas. Dengan sifat ini ResNet-50 mampu mengklasifikasikan citra sampah padat secara lebih efisien dan akurat (He *et al.*, 2016).

Penelitian Nnamoko *et al.* (2022) membangun model *5-layer Convolutional Neural Network* yang dilatih pada dua resolusi dataset klasifikasi sampah yang berisi 25.077 gambar, terdiri dari 13.966 gambar sampah organik dan 11.111 gambar sampah yang daur ulang. Hasil penelitian menunjukkan bahwa resolusi

gambar yang lebih kecil menghasilkan model yang lebih ringan, waktu pelatihan yang jauh lebih singkat, serta akurasi yang lebih baik dibandingkan model yang dilatih dengan resolusi gambar yang lebih besar. Selain itu, kedua model CNN yang diuji menunjukkan performa yang jauh lebih baik dibandingkan model *baseline random guess classifier* yang hanya mencapai akurasi 50,05%.

Penelitian lain oleh Feng *et al.* (2022) mengembangkan model klasifikasi sampah berbasis GECM-EfficientNet sebuah arsitektur *deep learning* yang merupakan modifikasi dari EfficientNetB0. Model ini disempurnakan dengan menyederhanakan modul *mobile inverted bottleneck convolution* (MBConv) serta menambahkan modul *efficient channel attention* (ECA) dan *coordinate attention* (CA) untuk meningkatkan efisiensi dan akurasi klasifikasi. Menggunakan dataset sampah rumah tangga yang mereka bangun sendiri serta dataset *TrashNet*, GECM-EfficientNet menghasilkan kinerja yang lebih baik dibandingkan arsitektur lain seperti EfficientNetB0, MobileNetv2, dan InceptionV3. Model GECM-EfficientNet yang dikembangkan memiliki jumlah parameter yang hanya 1,23 juta, serta memiliki waktu inferensi yang cepat yaitu 146ms per gambar, sehingga dapat digunakan untuk klasifikasi *real-time*. GECM-EfficientNet diterapkan pada tempat sampah pintar berbasis *Raspberry Pi 4B*, yang secara otomatis dapat mengenali, mengklasifikasikan, dan memilah sampah ke dalam kategori yang sesuai, seperti sampah daur ulang, sampah berbahaya, sampah dapur, dan sampah lainnya.

Penelitian ini bertujuan mengembangkan metode klasifikasi citra sampah padat dengan memanfaatkan model ResNet-50 yang dimana model ini dapat mengatasi beberapa kendala yang dihadapi dalam penelitian sebelumnya, seperti tingkat akurasi yang kurang optimal. Dengan menggunakan ResNet-50, penelitian ini diharapkan dapat mencapai akurasi yang lebih tinggi dalam klasifikasi sampah padat dan memberikan solusi yang lebih efisien dalam pemilahan dan daur ulang sampah padat.

1.2 Rumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah:

1. Bagaimana kinerja model ResNet-50 dalam mengklasifikasikan citra sampah padat?
2. Apakah model yang dikembangkan pada penelitian ini memberikan kinerja yang lebih baik dibanding penelitian sebelumnya?

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah:

1. Dataset yang digunakan dalam penelitian ini hanya mencakup gambar yang tidak sepenuhnya mewakili seluruh variasi dari data citra sampah padat yang ada.
2. Penelitian ini berfokus pada penggunaan model ResNet-50 dalam mengklasifikasikan dataset citra sampah padat, yang memerlukan penyesuaian parameter untuk memperoleh hasil yang optimal.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Menerapkan model *deep learning* dengan ResNet-50 untuk klasifikasi citra sampah padat.
2. Mengevaluasi kinerja dari model ResNet-50 yang telah dikembangkan untuk klasifikasi citra sampah padat.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

1. Menambah pengetahuan dalam bidang *deep learning* dan aplikasinya untuk klasifikasi citra sampah padat.

2. Berkontribusi dalam memperkaya pengetahuan tentang *deep learning* terutama mengenai ResNet-50 dan penggunaannya untuk klasifikasi citra digital.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian yang pernah dilakukan dan berkaitan dengan klasifikasi citra sampah padat menggunakan arsitektur ResNet-50 dapat dilihat pada Tabel 1.

Tabel 1. Penelitian terdahulu terkait klasifikasi sampah

No	Penelitian	Data	Metode	Hasil
1.	<i>Solid Waste Image Classification Using Deep Convolutional Neural Network</i> (Nnamoko et al., 2022)	Sampah Padat Jumlah : 24.705 Organik : 13.880 Daur Ulang : 10.825	Preprocessing <i>Resize, ImageDataGenerator.</i> Model <i>5-layer Convolutional Neural Network (CNN).</i>	Resolusi 225x264 Akurasi: 76,19% Resolusi 80x45 Akurasi: 80,88%

No	Penelitian	Data	Metode	Hasil
2.	<i>An Intelligent Waste-Sorting and Recycling Device Based on Improved EfficientNet</i> (Feng et al., 2022)	Sampah Rumah Tangga Jumlah: 7.361 Trashnet Jumlah: 2.527	Preprocessing <i>random size cropping, flipping, and luminance transformations.</i> Model <i>Transfer Learning</i> EfficientNetB0, GECM-EfficientNet.	Dataset Sampah Rumah Tangga Akurasi: 94,54% Dataset Trashnet Akurasi: 94,23
3.	<i>Waste Classification for Sustainable Development Using Image Recognition with Deep Learning Neural Network Models</i> (Malik et al., 2022)	Sampah Jumlah: 23.628	Preprocessing <i>Test-train-split method.</i> Model <i>Transfer Learning</i> EfficientNet-B0.	Akurasi: 85%
4.	Klasifikasi Jenis Sampah Menggunakan ResNet-50 (Nuariputri dkk., 2023)	Sampah Jumlah: 2751 citra 7 kelas: <i>cardboard, glass,metal, paper,plastic,trash, compost</i>	Preprocessing Augmentasi data <i>Split Validation.</i> Model ResNet-50 dimodifikasi <i>fine-tuning.</i>	Akurasi: 98.70%

Penelitian Nnamoko *et al.* (2022) mencoba untuk membangun arsitektur CNN yang akan digunakan klasifikasi citra sampah padat, dengan menggunakan dataset dari *kaggle* yang dirubah dan kemudian dibagi menjadi 2 resolusi gambar (80×45 dan 225×264 piksel) yang kemudian kedua dataset tersebut dibandingkan kinerja dan performanya. Didapatkan hasil bahwa Model resolusi kecil menghasilkan ukuran model yang lebih ringan, waktu pelatihan lebih singkat, dan akurasi lebih tinggi (80,88%) dibandingkan model resolusi besar (76,19%), meskipun model besar menunjukkan generalisasi yang lebih baik. Penelitian ini juga menyoroti pentingnya transparansi metodologi dan mempublikasikan setup eksperimental mereka untuk mendukung reproduksi penelitian.

Lalu penelitian Feng *et al.* (2022) mengembangkan model klasifikasi sampah berbasis GECM-EfficientNet, sebuah arsitektur *deep learning* ringan untuk mendukung perangkat pintar yang mampu menyortir dan mendaur ulang sampah secara otomatis. Dengan memanfaatkan dataset sampah rumah tangga yang dibangun sendiri dan dataset *TrashNet*, model ini mencapai akurasi 94,54% dan waktu inferensi 146ms pada perangkat *Raspberry Pi 4B* yang memenuhi kebutuhan inferensi *real-time*.

Penelitian Majchrowska *et al.* pada tahun 2022 mengembangkan kerangka kerja untuk klasifikasi sampah berbasis gambar menggunakan model *deep learning* EfficientNet-B0 yang diadaptasi melalui *transfer learning*. Pendekatan ini bertujuan meningkatkan efisiensi dan akurasi klasifikasi sampah padat rumah tangga. Model ini mencatat akurasi hingga 85%, dengan kebutuhan sumber daya komputasi yang jauh lebih rendah dibandingkan model sebelumnya seperti EfficientNet-B3, menjadikannya solusi yang bersaing.

Penelitian Nuariputri dkk. (2023) menggunakan model ResNet-50 sebagai basis untuk melakukan klasifikasi jenis sampah. Model ResNet-50 tersebut dimodifikasi dengan penambahan beberapa *layer* dan dilakukan *fine-tuning*.

Dataset yang digunakan terdiri dari 2751 citra yang terbagi menjadi 7 kelas, yaitu *cardboard*, *glass*, *metal*, *paper*, *plastic*, *trash*, dan *compost*, yang diperoleh dari situs *Kaggle*. Tahap *preprocessing* yang dilakukan meliputi augmentasi data dengan teknik *shearing*, *zooming*, dan *shifting*, serta pembagian data menggunakan *Split Validation*. Proses klasifikasi dijalankan menggunakan *Google Colab* dengan *library Tensorflow Keras*. Dengan parameter *batch size* 16, jumlah *epoch* 18, *learning rate* 0.0001, dan *optimizer Adam*, penelitian ini berhasil mencapai nilai akurasi sebesar 98.70%. Perbedaan mendasar dengan penelitian ini adalah penggunaan *ImageDataGenerator* sebagai teknik augmentasi dan dataset yang berbeda.

2.2 Sampah

Sampah seperti Gambar 1 adalah material yang dibuang karena dianggap tidak memiliki nilai oleh pemiliknya. Sampah padat seperti plastik, kaca, logam, kertas, dan bahan organik sering kali menjadi tantangan dalam pengelolaan lingkungan, terutama di kota besar yang menghasilkan sampah dalam jumlah besar setiap harinya (Malik *et al.*, 2022).

Sampah dapat dikelompokkan ke dalam beberapa kategori:

1. Sampah Organik: Sampah yang mudah terurai, seperti sisa makanan dan sampah taman (Nasir and Aziz Al-Talib, 2023).
2. Sampah Anorganik: Sampah yang tidak mengandung senyawa organik, seperti plastik, logam, dan kaca (Nasir and Aziz Al-Talib, 2023).



Gambar 1. Contoh sampah (Borut, 2022).

2.2.1 Sampah Padat

Sampah padat adalah limbah non-cair yang dihasilkan dari aktivitas rumah tangga, komersial, industri, dan institusi, serta terdiri dari bahan-bahan seperti kertas, plastik, logam, kaca, dan bahan organik. Berbeda dari limbah cair, sampah padat memerlukan penanganan fisik yang lebih kompleks dan memiliki volume yang lebih besar, sehingga menjadi fokus utama dalam strategi pengelolaan limbah perkotaan (Guerrero *et al.*, 2013).

Jika tidak dikelola dengan baik, sampah padat dapat menyebabkan pencemaran tanah, air, dan udara, serta menimbulkan dampak negatif terhadap kesehatan masyarakat. Peningkatan jumlah penduduk dan urbanisasi telah mendorong lonjakan volume sampah padat global. *World Bank* mencatat bahwa pada tahun 2020, produksi sampah padat dunia mencapai sekitar 2,01 miliar ton per tahun dan diperkirakan akan meningkat menjadi 3,4 miliar ton per tahun pada 2050 (Kaza *et al.*, 2018).

Dalam konteks modern, penggunaan teknologi berbasis kecerdasan buatan, seperti *deep learning* untuk klasifikasi otomatis sampah padat, menjadi solusi yang menjanjikan untuk meningkatkan efisiensi pengelolaan dan pemilahan sampah secara *real-time*.

2.3 Klasifikasi Gambar

Klasifikasi citra atau gambar adalah proses untuk mengelompokkan gambar ke dalam kelas-kelas berdasarkan pola tertentu. Teknik ini banyak digunakan di bidang seperti pengenalan wajah, pengelolaan data medis, dan pengklasifikasian objek.

Dalam pendekatan tradisional, proses klasifikasi melibatkan dua tahap utama, yaitu ekstraksi fitur dan klasifikasi. Fitur dari gambar, seperti warna, tekstur,

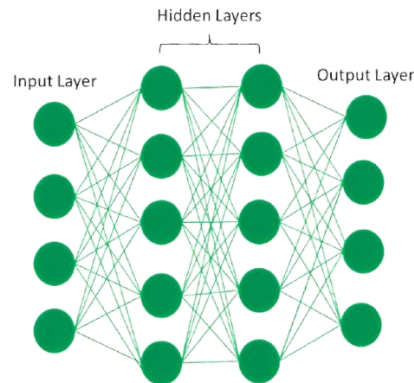
atau bentuk, digunakan untuk melatih algoritma klasifikasi yang kemudian dapat menentukan kategori suatu gambar (Medjahed, 2015).

2.4 *Machine Learning*

Machine learning (ML) adalah cabang dari kecerdasan buatan yang berfokus pada pengembangan algoritma yang memungkinkan komputer belajar dari data tanpa pemrograman eksplisit. ML digunakan dalam berbagai aplikasi seperti prediksi perilaku pelanggan, analisis genom, dan deteksi pola dalam data besar (Libbrecht and Noble, 2015). Teknologi ini telah menjadi pilar penting dalam komputasi modern, memanfaatkan metode statistik untuk memberikan keputusan berbasis data secara otomatis (Dietterich, 1996).

2.5 *Deep Learning*

Deep learning (DL) adalah subset dari *machine learning* yang menggunakan jaringan saraf tiruan (*neural networks*) dengan banyak lapisan (*deep neural networks*) untuk mempelajari representasi data yang kompleks. Tidak seperti algoritma *machine learning* tradisional, *deep learning* mampu mengekstrak fitur secara otomatis dari data mentah tanpa memerlukan rekayasa fitur manual (Schmidhuber, 2015). *Deep learning* telah mengungguli algoritma *machine learning* tradisional dalam banyak tugas berkat kemampuan pembelajaran hierarkisnya yang mendalam (Deng and Yu, 2013). Gambar 2 menunjukkan bentuk paling sederhana *deep learning* terbagi menjadi beberapa hierarki, *input layer*, *hidden layer*, dan *output layer*.



Gambar 2. *Neural Network Layer* (GeeksforGeeks, 2025).

2.5.1 *Input Layer*

Input layer adalah titik awal dari jaringan neural dalam *deep learning*, bertugas menerima data mentah dari luar untuk diproses lebih lanjut. Lapisan ini tidak melakukan komputasi apapun selain meneruskan data ke lapisan berikutnya yang bisa dalam bentuk vektor fitur atau representasi numerik dari data asli seperti gambar, teks, atau sinyal suara (Ramesh, 2017). Representasi input yang baik di lapisan ini sangat penting untuk memastikan kinerja optimal pada lapisan-lapisan berikutnya.

2.5.2 *Hidden Layer*

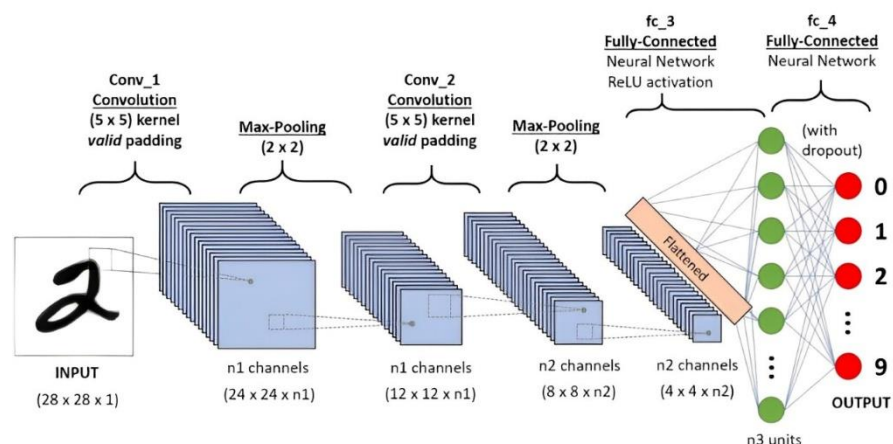
Hidden layer adalah inti komputasi dalam jaringan *neural* yang bertugas mengekstrak fitur dan pola dari data input. Lapisan ini terdiri dari *neuron-neuron* yang menerapkan fungsi aktivasi *non-linear* untuk mengubah data mentah menjadi representasi tingkat tinggi, memungkinkan jaringan untuk mempelajari hubungan kompleks dalam data (Bengio *et al.*, 2006). Semakin banyak *hidden layer* yang digunakan dalam arsitektur *deep learning*, semakin baik kemampuan jaringan untuk menangkap pola data yang kompleks, meskipun ini juga meningkatkan kebutuhan komputasi dan juga meningkatkan resiko *vanishing gradient*.

2.5.3 Output Layer

Output layer adalah lapisan terakhir dalam jaringan *neural* yang bertanggung jawab untuk menghasilkan prediksi atau keputusan akhir. Lapisan ini mengubah informasi yang telah diproses oleh *hidden layers* menjadi hasil yang dapat diinterpretasikan seperti probabilitas dalam klasifikasi atau nilai kontinu dalam regresi (He *et al.*, 2016). Jenis fungsi aktivasi yang digunakan di lapisan ini biasanya disesuaikan dengan tugas spesifik, seperti fungsi *linear* untuk regresi.

2.6 Convolutional Neural Network (CNN)

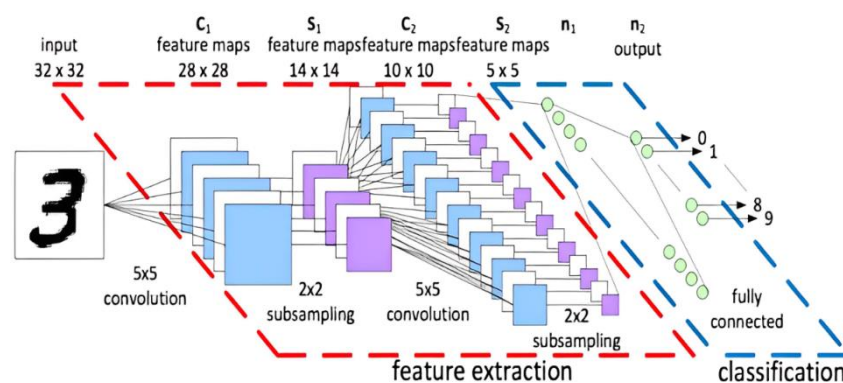
Convolutional Neural Networks (CNN) adalah salah satu arsitektur *deep learning* yang dirancang untuk memproses data yang memiliki *grid* pola, seperti gambar. CNN memanfaatkan operasi konvolusi untuk mengekstraksi fitur penting dari data masukan dan merupakan pilihan utama dalam berbagai aplikasi, termasuk klasifikasi gambar, pengenalan objek, dan pemrosesan bahasa alami (Albawi *et al.*, 2017). CNN terdiri dari beberapa jenis lapisan yang bekerja secara berurutan untuk mempelajari representasi data. Gambaran arsitektur CNN dapat dilihat pada Gambar 3.



Gambar 3. Struktur CNN (Dharmaraj, 2022).

2.6.1 Convolutional Layer

Lapisan konvolusi adalah inti dari CNN. Operasi konvolusi dilakukan dengan menggeser *kernel* (filter) pada data masukan untuk menghasilkan peta fitur (*feature map*). *Kernel* ini bertugas mendeteksi pola seperti tepi atau tekstur dalam gambar. Parameter pada *kernel* diperoleh melalui pelatihan model, yang memungkinkan CNN untuk belajar dari data (Frag, 2023). Proses ini dapat divisualisasikan dengan lebih jelas pada Gambar 4.



Gambar 4. *Convolutional Layer* (Bhatt et al., 2021).

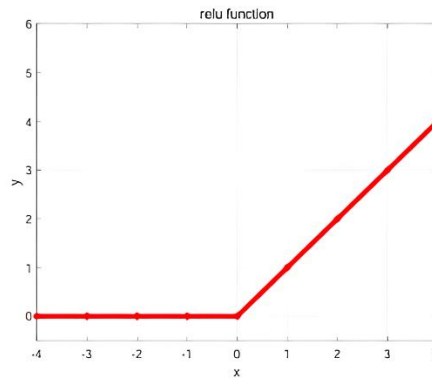
2.6.2 Activation Function

Fungsi aktivasi pada *Convolutional Neural Networks* (CNN) adalah sebuah fungsi matematis yang digunakan untuk menambahkan non-linearitas pada jaringan saraf. Tanpa fungsi aktivasi, jaringan saraf konvolusional akan hanya mampu mempelajari hubungan linear antar data, yang membatasi kemampuannya untuk memecahkan masalah yang kompleks, seperti pengenalan gambar atau klasifikasi objek.

Setelah operasi konvolusi, fungsi aktivasi seperti *Rectified Linear Unit* (*ReLU*) diterapkan untuk memperkenalkan non-linearitas. *ReLU*, misalnya, mengubah nilai negatif menjadi nol, sehingga mempercepat pelatihan dan mengurangi masalah *vanishing gradient* (Kuo, 2016).

2.6.2.1 *ReLU*

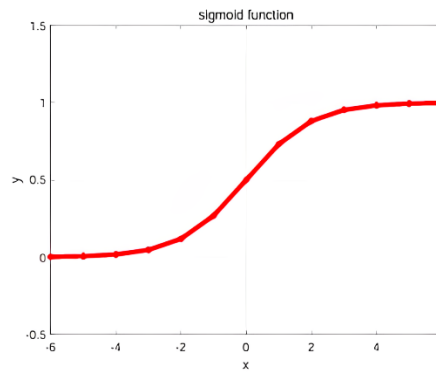
Fungsi aktivasi *ReLU* digunakan untuk memperkenalkan non-linearitas ke dalam CNN dengan memetakan nilai negatif menjadi nol dan mempertahankan nilai positif. Fungsi ini membantu mengatasi masalah *vanishing gradient*, sehingga mempercepat proses pelatihan dan meningkatkan efisiensi model secara keseluruhan (Jiang and Cheng, 2019). Namun, *ReLU* memiliki kekurangan seperti masalah "*dying ReLU*", di mana neuron menjadi tidak aktif sepenuhnya jika *output* terus-menerus bernilai negatif selama pelatihan (Wang *et al.*, 2020). Bentuk kurva fungsi *ReLU* ini dapat dilihat pada Gambar 5, yang menunjukkan bagaimana output *ReLU* tetap nol untuk input negatif dan meningkat secara linear untuk input positif.



Gambar 5. Ilustrasi *ReLU* (Wang *et al.*, 2020).

2.6.2.2 *Sigmoid*

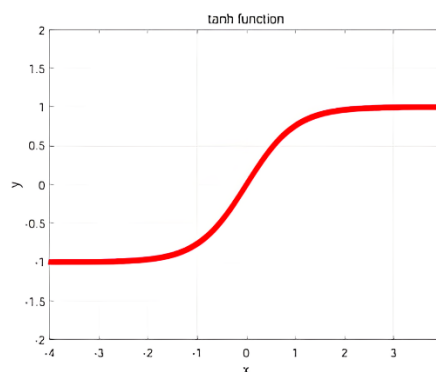
Fungsi aktivasi *Sigmoid* mengubah nilai input menjadi probabilitas antara 0 dan 1 sehingga sangat cocok untuk kasus yang membutuhkan keluaran probabilistik seperti klasifikasi biner. Fungsi ini menambahkan non-linearitas dan membantu dalam menyelesaikan tugas klasifikasi sederhana. Kelemahan utama fungsi ini adalah rentan terhadap masalah *vanishing gradient*, terutama ketika nilai input sangat besar atau kecil, yang dapat memperlambat konvergensi model (Zaheer and Shaziya, 2018). Bentuk kurva *sigmoid* dapat dilihat pada Gambar 6, yang memperlihatkan bagaimana output *sigmoid* meningkat secara bertahap dari 0 menuju 1 seiring bertambahnya nilai input.



Gambar 6. Ilustrasi *Sigmoid* (Wang *et al.*, 2020).

2.6.2.3 *Tanh*

Fungsi aktivasi *Tanh* adalah jenis fungsi aktivasi yang dikembangkan dari fungsi *Sigmoid*. *Tanh* memetakan nilai input ke dalam rentang -1 hingga 1, yang membuatnya lebih cocok untuk data yang memiliki nilai positif dan negatif. Fungsi ini membantu dalam menormalkan distribusi data pada *hidden layer*, yang meningkatkan stabilitas dan efisiensi pelatihan. Seperti *Sigmoid*, *Tanh* juga memiliki kelemahan berupa *vanishing gradient* pada nilai input yang ekstrim (Abdelouahab *et al.*, 2017). Bentuk kurva fungsi *tanh* dapat dilihat pada Gambar 7, yang menunjukkan bagaimana output meningkat secara gradual dari -1 ke 1 sesuai dengan nilai input.



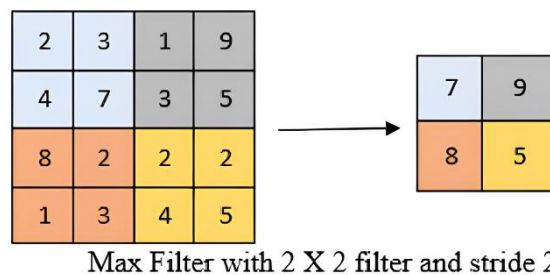
Gambar 7. Ilustrasi *Tanh* (Wang *et al.*, 2020).

2.6.3 Pooling Layer

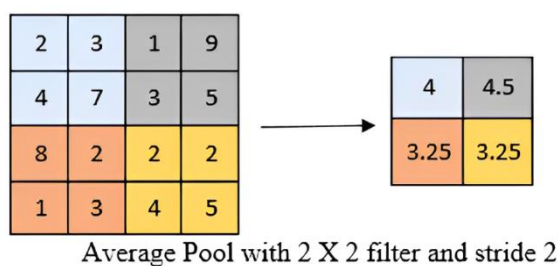
Pooling layer digunakan untuk mengurangi dimensi data dan kompleksitas komputasi, serta membuat model lebih tahan terhadap pergeseran dan distorsi kecil. Teknik *pooling* yang umum meliputi:

1. *Max Pooling*: Mengambil nilai maksimum dari area kecil pada peta fitur.
2. *Average Pooling*: Menghitung rata-rata nilai pada area pada peta fitur (Sun *et al.*, 2017).

Proses *Max Pooling* dapat dilihat pada Gambar 8, di mana *filter* 2×2 dengan *stride* 2 memilih nilai tertinggi di setiap wilayah, sementara itu Gambar 9 memperlihatkan proses *Average Pooling* dengan parameter yang sama, namun menghasilkan rata-rata dari setiap wilayah.



Gambar 8. *Max Pooling* (Bhatt *et al.*, 2021).

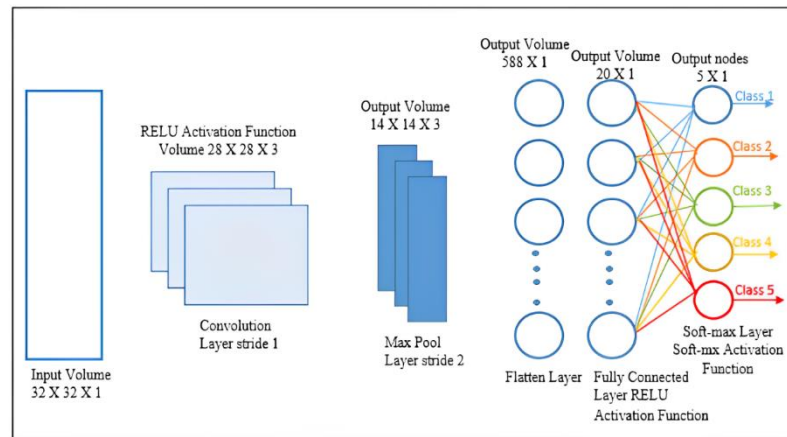


Gambar 9. *Average Pooling* (Bhatt *et al.*, 2021).

2.6.4 Fully Connected Layer

Lapisan ini terletak di bagian akhir jaringan dan menghubungkan semua *neuron* dari lapisan sebelumnya ke setiap *neuron* dalam lapisan ini. Fungsinya

adalah untuk melakukan klasifikasi berdasarkan fitur-fitur yang telah diekstraksi sebelumnya, proses ini dapat dilihat pada Gambar 10. Biasanya, lapisan ini diikuti oleh fungsi *softmax* atau fungsi aktivasi untuk menghasilkan probabilitas kelas (Krizhevsky *et al.*, 2017).



Gambar 10. *Fully Connected Layer* (Bhatt *et al.*, 2021).

2.6.5 Dropout Layer

Dropout adalah teknik regularisasi yang digunakan dalam *Convolutional Neural Networks* (CNN) untuk mencegah *overfitting*, yaitu kondisi di mana model belajar terlalu baik pada data pelatihan namun gagal menggeneralisasi pada data baru. *Dropout* bekerja dengan cara menonaktifkan secara acak sejumlah *neuron* selama pelatihan, sehingga mendorong jaringan untuk tidak bergantung terlalu besar pada *neuron* tertentu. Pada CNN, *dropout* umumnya diterapkan pada lapisan *fully connected*, namun penerapannya pada lapisan konvolusional juga terbukti meningkatkan generalisasi model (Park and Kwak, 2017). Penggunaan *dropout* dengan tingkat 30% (0.3) merupakan pilihan yang seimbang karena telah terbukti dalam studi empiris memberikan performa yang baik dalam mengatasi *overfitting* tanpa mengganggu proses pembelajaran secara signifikan (Peng, 2024). Selain itu, penelitian menunjukkan bahwa *dropout* serendah 10% pun dapat meningkatkan performa dibandingkan tanpa *dropout* sama sekali, dan 30% sering kali menjadi nilai yang kompromi antara efisiensi pelatihan dan akurasi model (Pauls and Yoder,

2018). Oleh karena itu, dalam skripsi ini digunakan *dropout* sebesar 30% untuk menjaga keseimbangan antara kemampuan generalisasi model dan kestabilan pelatihan.

2.7 *Vanishing Gradient*

Masalah *vanishing gradient* merupakan kendala umum dalam pelatihan jaringan saraf dalam, termasuk *Convolutional Neural Networks* (CNN). Masalah ini terjadi ketika gradien yang dihitung selama proses *backpropagation* menjadi sangat kecil, sehingga parameter di lapisan awal sulit diperbarui, yang menghambat konvergensi dan akurasi model secara keseluruhan (Sahu and Yadav, 2022). Untuk mengatasi masalah ini, skripsi ini menggunakan arsitektur ResNet-50, yang memanfaatkan *residual connections* (atau *skip connections*) yang memungkinkan gradien melewati beberapa lapisan secara langsung, sehingga mempercepat pelatihan dan memungkinkan jaringan menjadi lebih dalam tanpa mengalami degradasi performa (Sheikh *et al.*, 2024).

2.8 *Residual Network (ResNet)*

ResNet (*Residual Networks*) adalah arsitektur jaringan saraf konvolusional (CNN) yang diperkenalkan oleh He *et al.* pada tahun 2015. ResNet dirancang untuk mengatasi masalah *vanishing gradient* pada jaringan saraf yang lebih dalam, di mana semakin dalam lapisan jaringan semakin buruk kinerjanya meskipun jumlah parameternya meningkat. Pendekatan utama dalam ResNet adalah pengenalan *residual connections* atau *skip connections*, yang memungkinkan output dari suatu lapisan untuk diteruskan langsung ke lapisan lebih jauh tanpa perlu diproses melalui lapisan-lapisan di antaranya.

Keunggulan utama dari ResNet adalah kemampuannya untuk membangun jaringan yang sangat dalam tanpa kehilangan performa. Arsitektur ResNet telah digunakan secara luas dalam berbagai tugas *computer vision*, termasuk

deteksi objek, segmentasi citra, dan klasifikasi gambar, karena kemampuannya dalam menangkap fitur representatif yang mendalam dan kompleks. *Skip connection* pada ResNet memungkinkan propagasi gradien yang lebih stabil, sehingga mengurangi risiko *vanishing gradient* bahkan pada jaringan dengan ratusan lapisan.

Dalam penelitian He *et al.* (2016) terdapat beberapa versi dari ResNet itu sendiri, seperti ResNet-34, ResNet-50, ResNet-101, dan ResNet-152. Penelitian ini akan berfokus kepada arsitektur ResNet-50. Arsitektur ResNet-50 dapat dilihat pada Tabel 2.

Tabel 2. Arsitektur ResNet-50 (He *et al.*, 2016)

<i>Layer</i>	Ukuran Output	ResNet-50
<i>Conv_1</i>	112×112	7×7, 64, <i>stride</i> 2 3×3 <i>max pool</i> , <i>stride</i> 2
<i>Conv2_x</i>	56×56	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
<i>Conv3_x</i>	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
<i>Conv4_x</i>	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
<i>Conv5_x</i>	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
<i>FLOPs</i>		$3,8 \times 10^9$

Struktur ResNet-50, sebagaimana dirangkum dalam Tabel 2, terdiri dari beberapa blok utama:

1. **Conv_1**: Sebuah lapisan konvolusi awal (dengan filter 7x7, sejumlah 64 filter, dan *stride* 2), diikuti oleh *max pooling* (3x3 dengan *stride* 2) untuk mereduksi dimensi spasial input secara signifikan.

2. **Conv2_x, Conv3_x, Conv4_x, Conv5_x**: Ini adalah empat tahap utama dari blok residual. Setiap tahap terdiri dari beberapa blok residual. Blok residual dalam ResNet-50 biasanya menggunakan desain "*bottleneck*", yang terdiri dari tiga lapisan konvolusi:
 - a. Lapisan konvolusi 1x1 untuk mereduksi dimensi (jumlah filter).
 - b. Lapisan konvolusi 3x3 (lapisan utama yang melakukan pemrosesan fitur).
 - c. Lapisan konvolusi 1x1 untuk mengembalikan dimensi (jumlah filter) ke ukuran semula. Desain *bottleneck* ini lebih efisien secara komputasi dibandingkan dengan menggunakan dua lapisan konvolusi 3x3 secara langsung, terutama untuk jaringan yang lebih dalam. Jumlah blok residual dalam setiap tahap (misalnya, x3, x4, x6, x3 seperti yang ditunjukkan pada Tabel 2 untuk Conv2_x hingga Conv5_x) menentukan kedalaman total jaringan. Jumlah filter juga meningkat secara progresif pada setiap tahap (misalnya, dari 64 menjadi 256, lalu 512, 1024, dan akhirnya 2048 untuk *output* dari lapisan *bottleneck* terakhir).
3. **Global Average Pooling**: Setelah tahap konvolusi terakhir, biasanya diikuti oleh lapisan *Global Average Pooling* yang mereduksi setiap *feature map* menjadi satu nilai tunggal. Ini secara signifikan mengurangi jumlah parameter sebelum lapisan klasifikasi akhir.
4. **Fully Connected Layer**: Lapisan *output* berupa *fully connected layer* dengan jumlah neuron sesuai dengan jumlah kelas yang akan diprediksi (misalnya, 1000 untuk *ImageNet*, atau 2 dalam penelitian ini), diikuti oleh fungsi aktivasi *softmax*.

2.9 Ekstraksi Fitur

Ekstraksi fitur merupakan tahap krusial dalam pemrosesan citra yang bertujuan untuk mengubah data visual mentah menjadi representasi yang lebih sederhana namun tetap informatif, sehingga memudahkan proses analisis lanjutan. Tahap ini memungkinkan sistem untuk mengenali pola-pola penting dalam gambar

dengan cara menyoroti karakteristik seperti bentuk, tekstur, warna, atau tepi objek. Teknik ini sangat penting dalam berbagai aplikasi seperti klasifikasi objek, pengenalan wajah, segmentasi citra, serta pengelompokan data visual. Menurut Kumar *and* Bhatia (2014), proses ekstraksi fitur memungkinkan sistem pembelajaran mesin untuk bekerja lebih efisien dengan hanya memproses informasi yang relevan dari gambar, sehingga mengurangi kompleksitas komputasi dan meningkatkan akurasi dalam pengenalan pola (Kumar *and* Bhatia, 2014)

Ekstraksi fitur melibatkan proses mengambil karakteristik penting dari gambar yang dapat digunakan untuk analisis lebih lanjut. Teknik-teknik ini meliputi deteksi tepi, penghitungan *histogram of oriented gradients* (HOG), transformasi *wavelet*, dan metode berbasis pembelajaran mesin. Langkah-langkah awal sering melibatkan *preprocessing* gambar seperti binarisasi, normalisasi, dan reduksi dimensi (Kumar *and* Bhatia, 2014).

2.10 Dataset Splitting

Pembagian dataset (*dataset splitting*) adalah proses penting dalam pembelajaran mesin dan *deep learning* yang bertujuan untuk membagi kumpulan data menjadi beberapa subset terpisah agar model dapat dilatih, divalidasi, dan diuji secara efektif. Umumnya, dataset dibagi menjadi tiga bagian utama, yaitu data latih (*training set*) yang digunakan untuk melatih model, data validasi (*validation set*) yang digunakan untuk menyesuaikan parameter dan mencegah *overfitting*, serta data uji (*test set*) yang digunakan untuk mengukur kemampuan generalisasi model terhadap data baru. Menurut Goodfellow *et al.* (2016), pemisahan dataset ini penting untuk memastikan bahwa performa model tidak hanya baik pada data pelatihan, tetapi juga dapat diterapkan pada data yang belum pernah dilihat sebelumnya, yang mencerminkan kinerja nyata dari model tersebut (Goodfellow *et al.*, 2016)

Pemilihan proporsi pembagian data yang tepat sangat penting. Proporsi umum yang banyak digunakan adalah 60:20:20 atau 70:15:15, tergantung pada ukuran dataset dan tujuan eksperimen. Ketidakseimbangan dalam pembagian data dapat menyebabkan bias pada hasil pelatihan dan evaluasi terutama jika kelas dalam dataset tidak terdistribusi secara merata.

Prinsip penting lainnya adalah menjaga agar data uji benar-benar terpisah dari proses pelatihan dan validasi. Penggunaan data uji selama pelatihan dapat menyebabkan *information leakage*, yaitu kondisi di mana informasi dari data uji secara tidak sengaja memengaruhi parameter model, yang menyebabkan estimasi performa model menjadi terlalu optimis (Varma and Simon, 2006).

2.11 Augmentasi Data

Augmentasi data adalah teknik dalam pemrosesan data yang bertujuan meningkatkan keragaman dan jumlah data pelatihan dengan menerapkan transformasi pada data asli. Teknik ini umum digunakan dalam bidang *computer vision* untuk mengatasi keterbatasan data dan mengurangi risiko *overfitting* pada model *deep learning*. Transformasi yang diterapkan, seperti rotasi, pencerminan, *cropping*, atau perubahan kecerahan, menghasilkan variasi baru dari data yang ada tanpa mengubah makna dasarnya (Shorten and Khoshgoftaar, 2019).

Tujuan utama augmentasi data adalah meningkatkan generalisasi model dengan memperkenalkan variasi buatan, sehingga model menjadi lebih tahan terhadap perbedaan kondisi input saat diuji pada data asli. Selain itu, teknik ini juga membantu menangani ketidakseimbangan kelas (*class imbalance*) dan memaksimalkan pemanfaatan data yang tersedia, terutama dalam kasus di mana pengumpulan data baru sulit atau mahal (Perez and Wang, 2017).

2.11.1 *ImageDataGenerator*

Salah satu metode populer untuk menerapkan augmentasi data dalam kerangka kerja *TensorFlow/Keras* adalah *ImageDataGenerator*. Modul ini memungkinkan augmentasi gambar secara *real-time* selama pelatihan model dengan menghasilkan sampel gambar yang telah dimodifikasi melalui parameter transformasi yang ditentukan. Contoh parameter yang dapat diatur mencakup *rotation_range* (rotasi gambar), *width_shift_range* (geser horizontal), *zoom_range* (perbesaran), dan *horizontal_flip* (pencerminan horizontal) (Chollet, 2017).

Keunggulan *ImageDataGenerator* terletak pada kemampuannya menghasilkan data augmentasi secara dinamis tanpa memerlukan penyimpanan tambahan, sehingga efisien untuk dataset besar. Misalnya, dalam klasifikasi gambar medis, penggunaan kombinasi transformasi ini terbukti meningkatkan akurasi model hingga 15% dibandingkan pelatihan tanpa augmentasi (Perez and Wang, 2017).

2.12 *Confusion Matrix*

Confusion Matrix adalah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan prediksi yang dibuat oleh model terhadap label sebenarnya pada data uji (Amin, 2022). Matriks ini mencakup empat elemen utama dan contoh matriks ada pada Tabel 3:

1. *True Positive* (TP): Prediksi positif yang benar.
2. *False Positive* (FP): Prediksi positif yang salah (*Type I Error*).
3. *True Negative* (TN): Prediksi negatif yang benar.
4. *False Negative* (FN): Prediksi negatif yang salah (*Type II Error*).

Tabel 3. *Binary Confusion Matrix*

		Prediksi	
		Organik	Daur Ulang
Aktual	Organik	TP	FN
	Daur Ulang	FP	TN

2.12.1 Akurasi

Akurasi adalah metrik evaluasi yang digunakan untuk mengukur sejauh mana model dapat membuat prediksi yang benar secara keseluruhan terhadap total jumlah prediksi. Metrik ini mencerminkan proporsi data yang diklasifikasikan dengan benar, baik dalam kategori positif maupun negatif. Akurasi banyak digunakan dalam klasifikasi karena sederhana dan mudah dipahami, meskipun memiliki keterbatasan, terutama pada dataset yang tidak seimbang (Geng, 2024). Meskipun demikian, akurasi tetap menjadi salah satu metrik paling umum untuk mengevaluasi performa model klasifikasi karena dapat memberikan indikasi awal dari kinerja keseluruhan model (Dj Novakovi *et al.*, 2017). Persamaan akurasi dapat dilihat pada Persamaan 1.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots (1)$$

Meskipun mudah dipahami dan sering digunakan, akurasi memiliki keterbatasan pada dataset yang tidak seimbang (Chicco *and* Jurman, 2020), karena model dapat terlihat memiliki kinerja yang tinggi dengan hanya memprediksi kelas mayoritas secara dominan tanpa memperhatikan kelas minoritas, sehingga meskipun memberikan gambaran umum performa, akurasi kurang cocok untuk dataset yang tidak seimbang seperti yang telah disebutkan sebelumnya.

2.12.2 Presisi

Presisi mengukur proporsi prediksi positif yang benar-benar positif. Dengan kata lain, presisi menunjukkan seberapa baik model dalam menghindari kesalahan klasifikasi positif. Presisi tinggi berarti model jarang memberikan prediksi positif yang salah (Flach *and* Kull, 2015). Persamaan presisi dapat dilihat pada Persamaan 2.

$$Presisi = \frac{TP}{TP+FP} \dots\dots\dots (2)$$

2.12.3 F1-Score

F1-Score merupakan metrik evaluasi yang menggabungkan presisi dan sensitivitas (*recall*) dalam bentuk rata-rata harmonik, yang memberikan gambaran seimbang antara kedua metrik tersebut. Metrik ini sangat berguna dalam konteks dataset yang tidak seimbang, karena mampu menangkap kinerja model terhadap kelas minoritas yang sering kali diabaikan oleh metrik seperti akurasi (Riyanto *et al.*, 2023). Persamaan *F1-Score* dapat dilihat pada Persamaan 3.

$$F1-Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \dots\dots\dots (3)$$

2.12.4 Sensitivitas

Sensitivitas (*Recall*) atau dapat disebut *true positive rate* mengukur kemampuan suatu model untuk mendeteksi kasus yang benar-benar memiliki kondisi tertentu (positif), Sensitivitas digunakan untuk menilai seberapa andal model dalam menghindari kesalahan *false negative*, yakni saat kondisi positif gagal terdeteksi (van Iterson *et al.*, 2012). Persamaan Sensitivitas dapat dilihat pada Persamaan 4.

$$Sensitivitas = \frac{TP}{TP+FN} \dots\dots\dots (4)$$

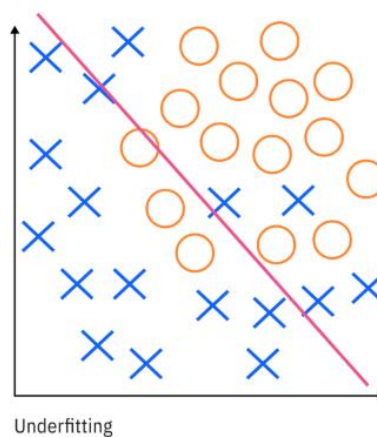
2.12.5 Spesifisitas

Spesifisitas, atau *true negative rate*, adalah metrik evaluasi yang menunjukkan kemampuan model klasifikasi untuk secara akurat mengidentifikasi kasus yang benar-benar negatif. Artinya, metrik ini mengukur proporsi data negatif yang berhasil dikenali sebagai negatif oleh model. Spesifisitas menjadi sangat penting dalam aplikasi-aplikasi yang menuntut minimnya kesalahan *false positive*, seperti sistem deteksi penyakit atau keamanan jaringan. Dalam evaluasi kinerja sistem klasifikasi, spesifisitas digunakan untuk menilai keandalan model dalam menghindari kesalahan klasifikasi kelas negatif (Sachdeva and Ali, 2022). Persamaan spesifisitas dapat dilihat pada Persamaan 5.

$$\text{Spesifisitas} = \frac{TN}{TN+FP} \dots\dots\dots (5)$$

2.13 Underfitting

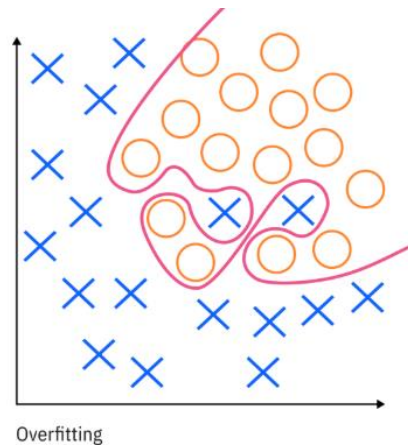
Underfitting adalah kondisi dimana model tidak mampu menangkap pola mendasar pada data latih, sehingga menghasilkan model yang memiliki kinerja yang kurang baik pada tahap *training* maupun *testing*. *Underfitting* dapat terjadi dikarenakan model terlalu sederhana untuk mempelajari pola dari data ataupun jumlah iterasi pelatihan yang kurang mencukupi (Zhang *et al.*, 2019). Ilustrasi *underfitting* dapat dilihat pada Gambar 11.



Gambar 11. Ilustrasi *underfitting* (IBM, 2024).

2.14 Overfitting

Overfitting adalah kondisi dimana model yang terlalu belajar pada data latih sehingga tidak memiliki kemampuan untuk menggeneralisasi ke data baru. *Overfitting* dapat terjadi dikarenakan model yang terlalu kompleks, atau dataset yang terbatas dari segi jumlah maupun variasi datanya, sehingga, kinerja pada data validasi atau data uji menurun drastis (Srivastava *et al.*, 2014). Salah satu upaya yang dapat dilakukan untuk mengurangi resiko *overfitting* adalah dengan melakukan augmentasi data (Ying, 2019). Ilustrasi *overfitting* dapat dilihat pada Gambar 12.



Gambar 12. Ilustrasi *overfitting* (IBM, 2024).

III. METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penjelasan mengenai tempat dan waktu penelitian adalah sebagai berikut.

3.1.1 Tempat Penelitian

Penelitian dilaksanakan di Laboratorium Rekayasa Perangkat Lunak (RPL), Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung yang beralamat di Jalan Prof. Dr. Soemantri Brodjonegoro No.1, Gedong Meneng, Kota Bandar Lampung, Lampung 35144.

3.1.2 Waktu Penelitian

Penelitian dilaksanakan pada semester ganjil tahun akademik 2024/2025, dimulai dari bulan November 2024. Untuk penjelasan kegiatannya terdapat pada Tabel 4.

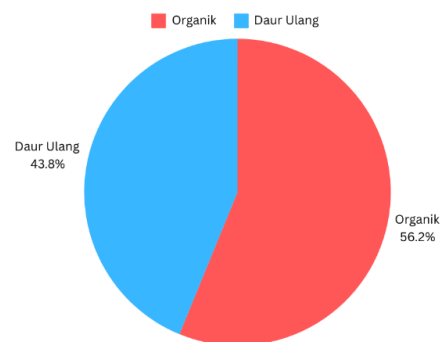
3.2 Data dan Alat

3.2.1 Data

Dataset yang digunakan dalam penelitian ini diperoleh dari penelitian Nnamoko *et al.* (2022). Data ini terdiri dari 24.705 data yang terbagi menjadi 13.880 data sampah organik dan 10.825 data sampah daur ulang. Dataset tersebut merupakan dataset turunan yang dikumpulkan oleh Sekar (2018). Contoh data dapat dilihat pada Gambar 13, serta rasio antara data organik dan data daur ulang ditampilkan pada Gambar 14.



Gambar 13. Contoh data *organic* (a) dan contoh data *recyclable* (b).



Gambar 14. Rasio data *organic* dan *recyclable*.

3.2.2 Alat

Berikut alat yang digunakan dalam penelitian ini:

1. Perangkat Keras (*Hardware*)

Perangkat keras berupa laptop yang digunakan dengan spesifikasi:

- a. Merk : Advan
- b. Model : Workpro
- c. *CPU* : Intel Core i5-1035G7
- d. *GPU* : Intel Iris Plus
- e. Ram : 16GB

2. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini:

- a. Sistem operasi *Windows* 11 64-bit versi 23H2
- b. Python 3.10.12

Python adalah bahasa pemrograman tingkat tinggi yang serbaguna dan mudah dipelajari, terkenal karena sintaksisnya yang bersih dan keterbacaan kode yang sangat baik. Bahasa ini mendukung berbagai paradigma pemrograman seperti imperatif, berorientasi objek, dan fungsional, menjadikannya pilihan populer di berbagai bidang, termasuk pengembangan web, ilmu data, dan kecerdasan buatan (Dierbach, 2014; Pandey, 2025).

- c. *Library Tensorflow* 2.17.1

TensorFlow adalah sebuah *library open-source* yang dirancang untuk komputasi numerik intensif, terutama dalam pengembangan model machine learning dan *deep learning*. *TensorFlow* memanfaatkan arsitektur fleksibel berbasis graf data yang dapat dijalankan di CPU, GPU, bahkan TPU, baik dalam perangkat lokal maupun skala kluster besar. Keunggulannya mencakup kemudahan pembangunan model, skalabilitas, dan dukungan luas terhadap berbagai algoritma pembelajaran mendalam (Prakash *et al.*, 2021).

d. *Library NumPy 1.26.4*

NumPy (Numerical Python) adalah pustaka fundamental dalam ekosistem Python untuk komputasi ilmiah, terutama digunakan untuk manipulasi *array* multidimensi dan operasi numerik tingkat tinggi. *NumPy* memungkinkan pemrograman *array* yang efisien dan cepat melalui struktur *ndarray* yang mendukung operasi vektor, matriks, serta integrasi dengan pustaka lain seperti *SciPy*, *Pandas*, dan *TensorFlow* (Harris *et al.*, 2020).

e. *Library Pandas 2.2.2*

Pandas adalah pustaka Python yang dirancang khusus untuk manipulasi dan analisis data terstruktur, menyediakan struktur data seperti *Series* dan *DataFrame* yang sangat efisien untuk pemrosesan, agregasi, dan pembersihan data. Pustaka ini banyak digunakan dalam berbagai bidang termasuk bisnis, riset ilmiah, dan industri karena kemampuannya menangani data skala besar secara fleksibel dan efisien (Enache, 2019).

f. *Library Matplotlib 3.8.0*

Matplotlib adalah pustaka visualisasi data 2D yang paling umum digunakan di Python, memungkinkan pengguna untuk membuat grafik seperti garis, batang, histogram, *scatter plot*, dan lainnya dengan kualitas publikasi. *Library* ini terkenal karena fleksibilitasnya dan kemampuannya menghasilkan visualisasi yang sangat dapat disesuaikan, baik untuk keperluan eksplorasi data maupun presentasi (Saini and Kumar, 2019).

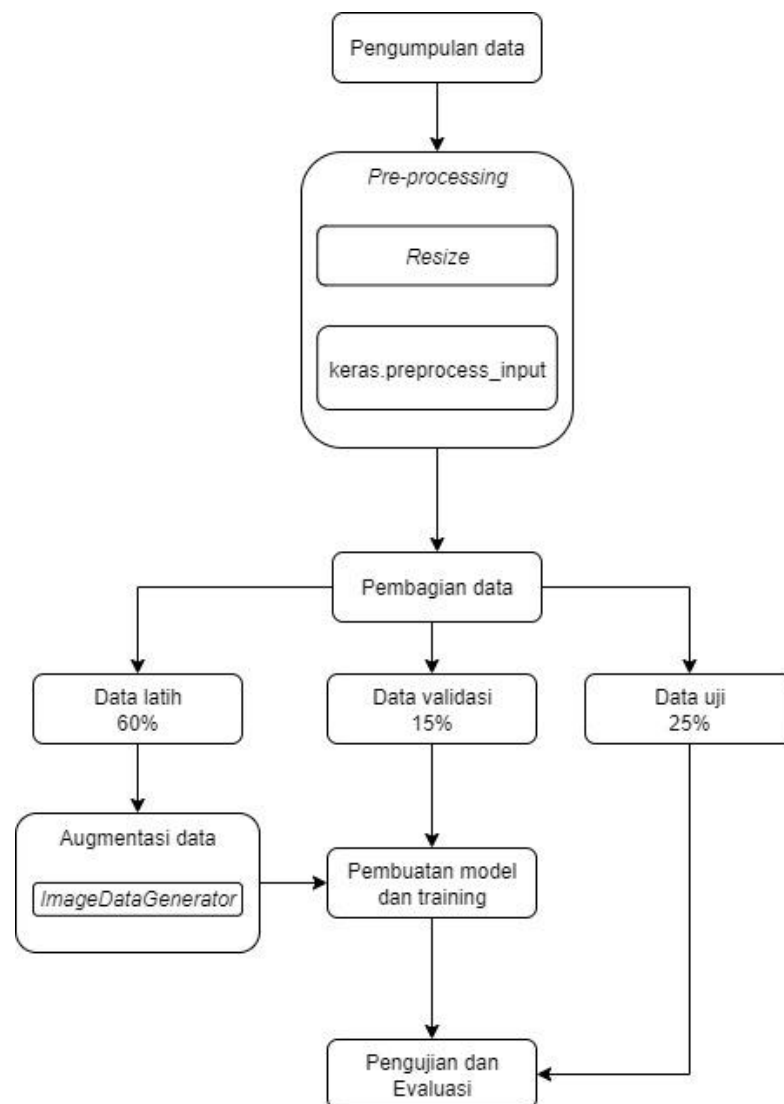
g. *Library Scikit-learn 1.6.0*

Scikit-learn adalah pustaka *open-source* berbasis Python yang menyediakan berbagai algoritma *machine learning* untuk klasifikasi, regresi, *clustering*, reduksi dimensi, dan validasi model. Dirancang dengan antarmuka yang sederhana dan konsisten, *Scikit-learn* sangat cocok bagi pemula maupun profesional, dan sering digunakan dalam pendidikan serta

industri karena kemudahan integrasi dengan pustaka Python lainnya seperti *NumPy* dan *Pandas* (Pedregosa *et al.*, 2011).

3.3 Tahap Penelitian

Tahapan penelitian yang akan dilakukan pada penelitian ini Sebagian besar mengikuti penelitian terdahulu oleh Nnamoko *et. al.* (2022), dimulai dari pengumpulan data, *preprocessing* data, augmentasi data, pembagian data, pembuatan model, hingga pengujian model. Tahapan selengkapnya dapat dilihat di Gambar 15.



Gambar 15. Tahapan Penelitian (Nnamoko *et. al.*, 2022).

3.3.1 Pengumpulan Data

Tahapan pertama pada penelitian ini adalah melakukan pengumpulan data. Pengumpulan data dilakukan dengan mengambil dataset yang telah digunakan oleh penelitian sebelumnya (Nnamoko *et al.*, 2022), yang dapat diakses pada tautan <https://data.mendeley.com/datasets/n3gtgm9jxj/3>. Dataset tersebut terdiri dari 24.705 gambar yang terbagi menjadi 2 kelas.

3.3.2 Pre-Processing

Preprocessing perlu dilakukan untuk memperbaiki kualitas dataset dan dapat meningkatkan efisiensi dan efektifitas pada saat *training* model dilakukan. Pada tahapan *preprocessing* ini sendiri ada beberapa hal yang akan dilakukan, seperti menerapkan *resize* menjadi ukuran 224x224, dan juga mengolah dataset menggunakan fungsi *preprocess_input* dari *Library Keras*.

3.3.3 Pembagian Data

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

- Data Latih
Data latih adalah data yang digunakan pada tahap *training* model. Data latih pada penelitian ini di ambil secara acak dari dataset utama, dalam penelitian ini data latih terdiri dari 60% dari total dataset.
- Data Uji
Data uji merupakan data yang digunakan pada tahap pengujian model. Data uji sendiri diambil dari 25% total dataset.
- Data Validasi
Data validasi merupakan data yang digunakan pada tahap *training* model untuk melakukan validasi terhadap hasil *training* di setiap

iterasinya. Data validasi yang digunakan pada penelitian ini diambil dari 15% total dataset.

3.3.4 Augmentasi Data

Augmentasi data dilakukan pada data latih untuk meningkatkan variasi dan total data pelatihan. Augmentasi diharapkan dapat membantu model menghindari *overfitting*. Teknik augmentasi yang akan digunakan pada penelitian ini adalah *ImageDataGenerator* yang disediakan oleh *library* Keras. Adapun Parameter yang akan digunakan pada penelitian ini dapat dilihat pada Tabel 5.

Tabel 5. Parameter augmentasi

Parameter	
<i>horizontalFlip</i>	<i>verticalFlip</i>
<i>rotation</i>	<i>widthShift</i>
<i>heightShift</i>	<i>shearRange</i>
<i>zoom</i>	<i>blur</i>
<i>brightness</i>	<i>contrast</i>
<i>saturation</i>	<i>hue</i>
<i>gamma</i>	

3.3.5 Pembuatan Model

Training model dalam penelitian ini dilakukan menggunakan model ResNet-50 dengan data yang akan dilatih sebanyak 207.522 citra sampah. *Hyperparameter* yang digunakan pada proses *training* dapat dilihat pada Tabel 6.

Tabel 6. *Hyperparameter training*

Parameter	Nilai
<i>Input size</i>	224 x 224
<i>Batch size</i>	64
<i>Optimizer</i>	<i>Adam</i>
<i>Epoch</i>	10
<i>Dropout</i>	0.3

3.3.6 Pengujian dan Evaluasi

Pengujian model dilakukan dengan memanfaatkan *confusion matrix*, yang dapat digunakan untuk menghitung beberapa metrik evaluasi, seperti akurasi, presisi, *F1-score*, sensitivitas dan spesitifitas, *confusion matrix* dapat memberikan detail tentang prediksi pada saat evaluasi sehingga memberikan gambaran akan performa model yang dibangun.

V. SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa:

1. Model ResNet-50 yang diterapkan dalam penelitian ini menunjukkan kinerja yang sangat baik untuk tugas klasifikasi citra sampah padat ke dalam dua kategori, yaitu *organic* dan *recyclable*. Model berhasil mencapai akurasi 95%, presisi 97% untuk kelas *organic* dan 93% untuk kelas *recyclable*, serta spesifisitas 96% untuk kelas *organic* dan 95% untuk kelas *recyclable*. Hasil ini mengindikasikan bahwa model mampu mengenali pola visual yang kompleks dengan tingkat kesalahan yang rendah. Keunggulan ini diperoleh berkat pemanfaatan arsitektur ResNet-50 yang memiliki *skip connection*, sehingga mampu mengatasi masalah *vanishing gradient* dan menjaga kestabilan pembelajaran pada jaringan yang sangat dalam. Selain itu, proses augmentasi data dan penggunaan teknik *transfer learning* turut meningkatkan kemampuan generalisasi model terhadap data baru.
2. Perbandingan dengan penelitian sebelumnya menunjukkan bahwa model ResNet-50 yang dikembangkan pada penelitian ini lebih unggul dibandingkan model CNN 5-layer milik Nnamoko et al. (2022) yang mencapai akurasi 80,88%, dan performanya sebanding dengan model GECCM-EfficientNet yang dikembangkan oleh Feng et al. (2022) dengan akurasi 94,54%. Hal ini menunjukkan bahwa penggunaan ResNet-50 sebagai arsitektur utama, dengan dukungan *preprocessing*,

augmentasi, dan *fine-tuning*, mampu menghasilkan model klasifikasi yang cukup akurat.

5.2 Saran

Untuk pengembangan penelitian selanjutnya, ada beberapa saran yang dapat dipertimbangkan:

1. Perluasan dataset dengan menambahkan variasi sampah padat dari lingkungan nyata (misalnya sampah campuran, logam, atau kaca) untuk meningkatkan generalisasi model.
2. Eksplorasi teknik augmentasi yang lebih spesifik, seperti *GAN-based* augmentation atau sintesis gambar berbasis *diffusion models*, untuk menangani kasus sampah dengan tekstur tidak biasa.
3. Implementasi *real-time* pada perangkat *IoT* atau robot pemilah sampah dengan optimasi *lightweight* model (misalnya ResNet-50 *quantized*) untuk mengurangi kebutuhan komputasi.

DAFTAR PUSTAKA

- Abdelouahab, K., Pelcat, M., & Berry, F. (2017). PhD forum: Why TanH is a hardware friendly activation function for CNNs. *ACM International Conference Proceeding Series, Part F1322*, 199–201. <https://doi.org/10.1145/3131885.3131937>
- Ahmed Medjahed, S. (2015). A Comparative Study of Feature Extraction Methods in Images Classification. *International Journal of Image, Graphics and Signal Processing*, 7(3), 16–23. <https://doi.org/10.5815/ijigsp.2015.03.03>
- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017, 2018-Janua*, 1–6. <https://doi.org/10.1109/ICEngTechnol.2017.8308186>
- Amin, M. F. (2022). Confusion Matrix in Binary Classification Problems: A Step-by-Step Tutorial. *Journal of Engineering Research*, 6(5), 0–0. <https://doi.org/10.21608/erjeng.2022.274526>
- Andrade, S. A., & Mondelli, G. (2024). Assessing of Manual Sorting in Municipal Solid Waste: A Study Case in São Paulo City. *Revista Gestão & Sustentabilidade*, 5(1), e14344–e14344. <https://doi.org/10.36661/2596-142x.2023v5n1.14344>
- Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2006). Greedy Layer-Wise Training of Deep Networks. In *NIPS 2006: Proceedings of the 19th International Conference on Neural Information Processing Systems* (pp. 153–160). The MIT Press. <https://doi.org/10.7551/mitpress/7503.003.0024>
- Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). Cnn variants for computer vision: History, architecture, application, challenges and future scope. *Electronics (Switzerland)*, 10(20), 2470. <https://doi.org/10.3390/electronics10202470>
- Borut. (2022). *(III) What Is the Essence of Detoxification?* Medium. <https://medium.com/illumination-curated/iii-what-is-the-essence-of-detoxification-60d47cc52245>
- Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1), 6. <https://doi.org/10.1186/s12864-019-6413-7>

- Chollet, F. (2017). *Deep Learning with Python*. Manning Publications. <https://consensus.app/papers/deep-learning-with-python-chollet/7d40941d70a05f559057a4e2e990d12a/>
- Deng, L., & Yu, D. (2013). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4), 197–387. <https://doi.org/10.1561/20000000039>
- Dharmaraj. (2022). *Convolutional Neural Networks (CNN) — Architecture Explained*. Medium. <https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243>
- Dierbach, C. (2014). Python as a first programming language. *J. Comput. Sci. Coll.*, 29(6), 153–154.
- Dietterich, T. (1996). Machine learning. *ACM Computing Surveys*, 28(4es), 3. <https://doi.org/10.1145/242224.242229>
- Dj Novakovi, J., Veljovi, A., Ili, S. S., Zeljko Papi, ~, & Tomovi, M. (2017). Evaluation of Classification Models in Machine Learning. *Theory and Applications of Mathematics & Computer Science*, 7(1), 39–46. <https://api.semanticscholar.org/CorpusID:125586327>
- Emmatty, F. J., & Panicker, V. V. (2022). Workplace-based assessment and intervention design for waste sorting tasks in a developing country. *Sadhana - Academy Proceedings in Engineering Sciences*, 47(1), 40. <https://doi.org/10.1007/s12046-022-01804-7>
- Enache, M. C. (2019). Data Analysis with Pandas. *Annals of Dunarea de Jos University of Galati. Fascicle I. Economics and Applied Informatics*, 25(2), 69–74. <https://doi.org/10.35219/eai1584040933>
- Farag, M. M. (2023). Design and Analysis of Convolutional Neural Layers: A Signal Processing Perspective. *IEEE Access*, 11, 27641–27661. <https://doi.org/10.1109/ACCESS.2023.3258399>
- Feng, Z., Yang, J., Chen, L., Chen, Z., & Li, L. (2022). An Intelligent Waste-Sorting and Recycling Device Based on Improved EfficientNet. *International Journal of Environmental Research and Public Health*, 19(23), 15987. <https://doi.org/10.3390/ijerph192315987>
- Flach, P., & Kull, M. (2015). Precision-Recall-Gain Curves: PR Analysis Done Right. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 28). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2015/file/33e8075e9970de0cfea955afd4644bb2-Paper.pdf
- GeeksforGeeks. (2025). *Introduction to Deep Learning*. GeeksforGeeks. <https://www.geeksforgeeks.org/introduction-deep-learning/>
- Geng, S. (2024). Analysis of the Different Statistical Metrics in Machine Learning. *Highlights in Science, Engineering and Technology*, 88, 350–356.

<https://doi.org/10.54097/jhq3tv19>

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Guerrero, L. A., Maas, G., & Hogland, W. (2013). Solid waste management challenges for cities in developing countries. *Waste Management*, 33(1), 220–232. <https://doi.org/10.1016/j.wasman.2012.09.008>
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- IBM. (2024). *Overfitting vs. underfitting: Finding the balance*. IBM. <https://www.ibm.com/think/topics/overfitting-vs-underfitting>
- Jiang, T., & Cheng, J. (2019). Target Recognition Based on CNN with LeakyReLU and PReLU Activation Functions. *Proceedings - 2019 International Conference on Sensing, Diagnostics, Prognostics, and Control, SDPC 2019*, 718–722. <https://doi.org/10.1109/SDPC.2019.00136>
- Kaza, S., Yao, L. C., Bhada-Tata, P., & Van Woerden, F. (2018). What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050. In *What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050*. Washington, DC: World Bank. <https://doi.org/10.1596/978-1-4648-1329-0>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. <https://doi.org/10.1145/3065386>
- Kumar, G., & Bhatia, P. K. (2014). A detailed review of feature extraction in image processing systems. *International Conference on Advanced Computing and Communication Technologies, ACCT*, 5–12. <https://doi.org/10.1109/ACCT.2014.74>
- Kuo, C. C. J. (2016). Understanding convolutional neural networks with a mathematical model. *Journal of Visual Communication and Image Representation*, 41, 406–413. <https://doi.org/10.1016/j.jvcir.2016.11.003>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Libbrecht, M. W., & Noble, W. S. (2015). Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6), 321–332. <https://doi.org/10.1038/nrg3920>
- Malik, M., Sharma, S., Uddin, M., Chen, C. L., Wu, C. M., Soni, P., & Chaudhary,

- S. (2022). Waste Classification for Sustainable Development Using Image Recognition with Deep Learning Neural Network Models. *Sustainability (Switzerland)*, 14(12), 7222. <https://doi.org/10.3390/su14127222>
- Nasir, I., & Aziz Al-Talib, G. A. (2023). Waste Classification Using Artificial Intelligence Techniques: Literature Review. *Technium: Romanian Journal of Applied Sciences and Technology*, 5, 49–59. <https://doi.org/10.47577/technium.v5i.8345>
- Nitish, K. (2023). *Exploring ResNet50: An In-Depth Look at the Model Architecture and Code Implementation*. Medium. <https://medium.com/@nitishkundu1993/exploring-resnet50-an-in-depth-look-at-the-model-architecture-and-code-implementation-d8d8fa67e46f>
- Nnamoko, N., Barrowclough, J., & Procter, J. (2022). Solid Waste Image Classification Using Deep Convolutional Neural Network. *Infrastructures*, 7(4), 47. <https://doi.org/10.3390/infrastructures7040047>
- Nuariputri, J., Maimunah, & Sukmasetya, P. (2023). Klasifikasi Jenis Sampah Menggunakan Base ResNet-50. *Jurnal Ilmiah Komputasi*, 22(3), 379–383. <https://doi.org/10.32409/jikstik.22.3.3380>
- Pandey, P. (2025). Python and Its Implications. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, 09(02), 1–9. <https://doi.org/10.55041/IJSREM41465>
- Park, S., & Kwak, N. (2017). Analysis on the dropout effect in convolutional neural networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 10112 LNCS* (pp. 189–204). https://doi.org/10.1007/978-3-319-54184-6_12
- Pauls, A., & Yoder, J. A. (2018). Determining Optimum Drop-out Rate for Neural Networks. *The Bridge - The Magazine of IEEE-Eta Kappa Nu*, 10–16. <https://api.semanticscholar.org/CorpusID:247595430>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85), 2825–2830. <http://jmlr.org/papers/v12/pedregosa11a.html>
- Peng, C. (2024). Comprehensive Analysis of the Impact of Learning Rate and Dropout Rate on the Performance of Convolutional Neural Networks on the CIFAR-10 Dataset. *Applied and Computational Engineering*, 102(1), 183–192. <https://doi.org/10.54254/2755-2721/102/20241161>
- Perez, L., & Wang, J. (2017). *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. <https://arxiv.org/abs/1712.04621>
- Prakash, K. B., Ruwali, A., & Kanagachidambaresan, G. R. (2021). Introduction to Tensorflow Package. In *EAI/Springer Innovations in Communication and Computing* (pp. 1–4). https://doi.org/10.1007/978-3-030-57077-4_1

- Ramesh, V. (2017). A Review on Application of Deep Learning in Thermography. *International Journal of Engineering and Management Research (IJEMR)*, 7, 489–493.
- Riyanto, S., Sitanggang, I. S., Djatna, T., & Atikah, T. D. (2023). Comparative Analysis using Various Performance Metrics in Imbalanced Data for Multi-class Text Classification. *International Journal of Advanced Computer Science and Applications*, 14(6), 1082–1090. <https://doi.org/10.14569/IJACSA.2023.01406116>
- Sachdeva, S., & Ali, A. (2022). Machine learning with digital forensics for attack classification in cloud network environment. *International Journal of System Assurance Engineering and Management*, 13(S1), 156–165. <https://doi.org/10.1007/s13198-021-01323-4>
- Sahu, S. K., & Yadav, R. N. (2022). Key facial points recognition using ResNet. *Materials Today: Proceedings*, 66(8), 3651–3656. <https://doi.org/10.1016/j.matpr.2022.07.342>
- Saini, M., & Kumar, R. (2019). Exploring Data Visualization and Analysis with Matplotlib. *International Journal of Psychosocial Rehabilitation*, 23(4), 2189–2194. <https://doi.org/https://doi.org/10.61841/y6yt6m46>
- Satav, A. G., Kubade, S., Amrutkar, C., Arya, G., & Pawar, A. (2023). A state-of-the-art review on robotics in waste sorting: scope and challenges. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 17(6), 2789–2806. <https://doi.org/10.1007/s12008-023-01320-w>
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- Sheikh, L. M. K., Shaikh, A., Sandupatla, A., Pudale, R., Bakare, A., & Chavan, P. M. (2024). Classification of Simple CNN Model and ResNet50. *International Journal for Research in Applied Science and Engineering Technology*, 12(4), 4606–4610. <https://doi.org/10.22214/ijraset.2024.60677>
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60. <https://doi.org/10.1186/s40537-019-0197-0>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15, 1929–1958. <https://doi.org/10.5555/2627435.2670313>
- Sun, M., Song, Z., Jiang, X., Pan, J., & Pang, Y. (2017). Learning Pooling for Convolutional Neural Network. *Neurocomputing*, 224, 96–104. <https://doi.org/10.1016/j.neucom.2016.10.049>
- van Iterson, M., van Haagen, H. H. H. B. M., & Goeman, J. J. (2012). Resolving confusion of tongues in statistics and machine learning: A primer for biologists and bioinformaticians. *PROTEOMICS*, 12(4–5), 543–549. <https://doi.org/10.1002/pmic.201100395>

- Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7(1), 91. <https://doi.org/10.1186/1471-2105-7-91>
- Wang, Y., Li, Y., Song, Y., & Rong, X. (2020). The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition. *Applied Sciences*, 10(5), 1897. <https://doi.org/10.3390/app10051897>
- Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, 1168, 22022. <https://doi.org/10.1088/1742-6596/1168/2/022022>
- Zaheer, R., & Shaziya, H. (2018). GPU-based empirical evaluation of activation functions in convolutional neural networks. *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, 769–773. <https://doi.org/10.1109/ICISC.2018.8398903>
- Zhang, H., Zhang, L., & Jiang, Y. (2019). Overfitting and Underfitting Analysis for Deep Learning Based End-to-end Communication Systems. *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, 1–6. <https://doi.org/10.1109/WCSP.2019.8927876>