EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer DOI: https://doi.org/10.47065/explorer.v6i2.2670

Optimasi Kinerja Model LeNet Berbasis Deep Learning untuk Klasifikasi Citra Menggunakan Pendekatan Hyperparameter Tuning

Nita Syahputri[1*], Ommi Alfina[2]
[1*,2] Program Studi Sistem Informasi, Fakultas Teknik dan ilmu Komputer, Universitas Pembinaan Masyarakat Indonesia, Medan, Indonesia
[1*]nieta20d@gmail.com, [2]ny.aroen@gmail.com
[*)] nieta20d@gmail.com

Abstrak-Perkembangan Deep Learning telah memberikan kontribusi signifikan dalam bidang klasifikasi citra, khususnya melalui penggunaan Convolutional Neural Network (CNN). Namun, salah satu permasalahan utama dalam implementasi model CNN adalah belum optimalnya kinerja model akibat pemilihan hyperparameter yang kurang tepat. Model sederhana seperti LeNet sering dianggap memiliki keterbatasan performa dibandingkan arsitektur modern, padahal dengan pendekatan yang tepat, model ini masih memiliki potensi untuk menghasilkan kinerja yang kompetitif. Oleh karena itu, penelitian ini bertujuan untuk meningkatkan kinerja model LeNet dalam klasifikasi citra melalui pendekatan hyperparameter tuning. Metode yang digunakan dalam penelitian ini meliputi tahap preprocessing data, pembagian dataset, implementasi model LeNet sebagai baseline, serta optimasi hyperparameter yang mencakup learning rate, batch size, optimizer, dan jumlah epoch. Dataset yang digunakan merupakan citra angka tulisan tangan yang telah melalui proses normalisasi dan transformasi agar sesuai dengan input model CNN. Evaluasi kinerja model dilakukan menggunakan metrik accuracy, precision, recall, dan F1-score. Hasil penelitian menunjukkan bahwa optimasi hyperparameter memberikan peningkatan performa yang signifikan terhadap model LeNet. Model baseline menghasilkan akurasi sebesar 97.52%, sedangkan model hasil optimasi terbaik mampu mencapai akurasi sebesar 98.93%. Selain itu, nilai precision, recall, dan F1-score juga mengalami peningkatan yang menunjukkan bahwa model memiliki kemampuan klasifikasi yang lebih baik dan seimbang. Dengan demikian, penelitian ini membuktikan bahwa optimasi hyperparameter merupakan pendekatan yang efektif untuk meningkatkan kinerja model CNN sederhana tanpa meningkatkan kompleksitas arsitektur.

Kata Kunci: Learning, Convolutional Neural Network, LeNet, Hyperparameter Tuning, Klasifikasi Citra

Abstract-The development of Deep Learning has made significant contributions to the field of image classification, particularly through the use of Convolutional Neural Networks (CNN). However, one of the main problems in implementing CNN models is the suboptimal performance of the model due to inappropriate hyperparameter selection. Simple models such as LeNet are often considered to have limited performance compared to modern architectures, even though with the right approach, this model still has the potential to produce competitive performance. Therefore, this study aims to improve the performance of the LeNet model in image classification through a hyperparameter tuning approach. The methods used in this study include data preprocessing, dataset division, implementation of the LeNet model as a baseline, and hyperparameter optimization including learning rate, batch size, optimizer, and number of epochs. The dataset used is a handwritten number image that has been normalized and transformed to match the CNN model input. Model performance evaluation was carried out using accuracy, precision, recall, and F1-score metrics. The results showed that hyperparameter optimization provided a significant performance improvement over the LeNet model. The baseline model produced an accuracy of 97.52%, while the best optimized model achieved an accuracy of 98.93%. Furthermore, precision, recall, and F1-score values also improved, indicating the model's improved and more balanced classification capabilities. Thus, this study demonstrates that hyperparameter optimization is an effective approach to improving the performance of simple CNN models without increasing architectural complexity.

Keywords: Deep Learning, Convolutional Neural Network, LeNet, Hyperparameter Tuning, Image Classification

1. PENDAHULUAN

Perkembangan teknologi Artificial Intelligence (AI) dalam beberapa dekade terakhir telah mengalami peningkatan yang sangat pesat, terutama pada cabang Deep Learning yang menjadi tulang punggung dalam berbagai aplikasi cerdas [1], [2], [3]. Salah satu bidang yang mengalami kemajuan signifikan adalah klasifikasi citra digital, yang banyak diterapkan dalam berbagai sektor seperti kesehatan, pertanian, keamanan, dan industri. Kemampuan sistem dalam mengklasifikasikan citra secara otomatis menjadi sangat penting untuk meningkatkan efisiensi dan akurasi dalam pengambilan keputusan berbasis data [4], [5], [6].

Pendekatan yang paling dominan dalam klasifikasi citra saat ini adalah penggunaan Convolutional Neural Network (CNN) [4], [7], [8], yang terbukti mampu mengekstraksi fitur secara hierarkis dari data citra [9], [10], [11]. CNN memiliki keunggulan dibandingkan metode tradisional karena tidak memerlukan proses ekstraksi fitur secara manual. Salah satu arsitektur awal CNN yang cukup terkenal adalah model LeNet, yang diperkenalkan sebagai solusi untuk pengenalan karakter tulisan tangan [12], [13], [14], [15]. Meskipun tergolong sebagai arsitektur klasik, LeNet masih relevan digunakan hingga saat ini sebagai model dasar (baseline) dalam berbagai penelitian, terutama pada dataset berukuran kecil hingga menengah. Pada penelitian terdahulu yang dilakukan oleh [16] didalam artikel ini memiliki kelebihan utama pada pendekatan sistematis dalam optimasi hyperparameter menggunakan Grid Search pada arsitektur CNN VGG16, yang didukung oleh dataset cukup besar (5.099 citra) serta tahapan preprocessing dan augmentasi yang jelas sehingga mampu menghasilkan performa sangat tinggi (akurasi 99,59% dengan precision, recall, dan F1-score 0,996), serta memberikan pipeline yang relatif terstruktur dan mudah direplikasi untuk klasifikasi penyakit buah delima ; namun demikian, kelemahannya terletak pada aspek novelty yang masih terbatas karena hanya mengombinasikan metode yang sudah umum (VGG16 + Grid Search), ruang eksplorasi hyperparameter yang sempit, tidak adanya perbandingan dengan model lain (misalnya ResNet atau EfficientNet), serta potensi bias karena hanya menggunakan dataset publik tanpa validasi pada data lapangan nyata sehingga generalisasi model masih diragukan, di samping kurangnya analisis mendalam terhadap kesalahan klasifikasi dan tidak adanya pembahasan terkait kompleksitas komputasi atau efisiensi model. Sedangkan pada penelitian yang dilakukan oleh [15] didalam artikel ini memiliki kelebihan pada penggunaan dataset yang relatif besar (lebih dari 25.000 citra) serta pendekatan komparatif antara dua arsitektur modern, yaitu EfficientNet-B0 dan ResNet-50, yang diperkuat dengan penerapan transfer learning, augmentasi data, serta optimasi hyperparameter menggunakan Keras Tuner sehingga menghasilkan performa yang cukup tinggi (akurasi hingga 97,25%) dan analisis yang komprehensif mencakup aspek akurasi, efisiensi komputasi, dan waktu inferensi ; selain itu, penelitian ini juga menunjukkan nilai praktis melalui potensi implementasi real-time dan analisis dampak augmentasi yang jelas, namun kelemahannya terletak pada kontribusi kebaruan (novelty) yang masih terbatas karena hanya membandingkan model yang sudah umum tanpa modifikasi arsitektur atau metode baru, ruang eksplorasi hyperparameter yang relatif sempit, serta tidak adanya pengujian pada dataset lain atau skenario multi-kelas yang lebih kompleks sehingga generalisasi model masih terbatas, di samping kurangnya analisis mendalam terkait error (misclassification), explainability model, dan perbandingan dengan pendekatan state-of-the-art terbaru seperti Vision Transformer atau hybrid model.

Namun demikian, seiring berkembangnya arsitektur CNN modern seperti VGG [17], [18], ResNet [19], [20], [21], dan DenseNet [22], [23], [24], model LeNet sering kali dianggap memiliki keterbatasan dalam hal performa, khususnya dalam mencapai tingkat akurasi yang tinggi pada permasalahan klasifikasi yang lebih kompleks [25], [26], [27]. Anggapan ini menyebabkan sebagian peneliti cenderung mengabaikan potensi model sederhana seperti LeNet, meskipun model tersebut memiliki keunggulan dari sisi efisiensi komputasi, kebutuhan memori yang rendah, serta kemudahan implementasi. Permasalahan utama yang sering dihadapi dalam penggunaan model CNN, termasuk LeNet, bukan hanya terletak pada desain arsitektur, tetapi juga pada proses pelatihan model yang sangat dipengaruhi oleh pemilihan hyperparameter [28], [29], [30]. Hyperparameter seperti learning rate, batch size, jumlah epoch, serta jenis optimizer memainkan peran krusial dalam menentukan keberhasilan proses pembelajaran model. Pemilihan nilai hyperparameter yang tidak tepat dapat menyebabkan model mengalami underfitting, di mana model tidak mampu menangkap pola data dengan baik, atau overfitting, di mana model terlalu menyesuaikan diri dengan data pelatihan sehingga performanya menurun pada data pengujian [31], [32]. Dalam praktiknya, penentuan hyperparameter masih sering dilakukan secara manual melalui pendekatan trial and error, yang bersifat tidak sistematis dan memerlukan waktu yang cukup lama. Pendekatan ini juga tidak menjamin bahwa kombinasi hyperparameter yang diperoleh merupakan yang paling optimal. Oleh karena itu, diperlukan suatu pendekatan optimasi yang lebih terstruktur untuk mengeksplorasi berbagai kombinasi hyperparameter guna meningkatkan performa model secara signifikan [33], [34]. Sejumlah penelitian telah menunjukkan bahwa optimasi hyperparameter mampu memberikan peningkatan kinerja yang cukup signifikan pada model deep learning. Namun, sebagian besar penelitian tersebut berfokus pada arsitektur yang kompleks dan berukuran besar, sehingga membutuhkan sumber daya komputasi yang tinggi. Di sisi lain, penelitian yang mengkaji optimasi pada arsitektur sederhana seperti LeNet masih relatif terbatas. Padahal, dalam banyak kasus praktis, terutama pada lingkungan dengan keterbatasan perangkat keras, model yang ringan namun tetap memiliki performa yang baik menjadi sangat dibutuhkan [35], [36], [37], [38]. Selain itu, penting untuk memahami bahwa peningkatan performa model tidak selalu harus dilakukan dengan menambah kompleksitas arsitektur. Pendekatan alternatif yang lebih efisien adalah dengan melakukan optimasi terhadap parameter pelatihan yang digunakan. Dengan strategi yang tepat, model sederhana seperti LeNet berpotensi menghasilkan performa yang kompetitif tanpa memerlukan biaya komputasi yang tinggi.

Berdasarkan uraian tersebut, penelitian ini berfokus pada upaya untuk mengoptimalkan kinerja model LeNet dalam klasifikasi citra melalui pendekatan hyperparameter tuning. Proses optimasi dilakukan dengan menguji

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer DOI: https://doi.org/10.47065/explorer.v6i2.2670

berbagai kombinasi hyperparameter seperti learning rate, batch size, jumlah epoch, dan jenis optimizer, untuk menemukan konfigurasi terbaik yang mampu meningkatkan akurasi dan stabilitas model. Selain itu, penelitian ini juga melakukan analisis perbandingan antara performa model LeNet sebelum dan sesudah dilakukan optimasi, sehingga dapat memberikan gambaran yang jelas mengenai dampak dari proses tuning yang dilakukan. Dengan dilakukannya penelitian ini, tentu dapat memberikan kontribusi dalam pengembangan metode klasifikasi citra yang lebih efisien dan mudah diimplementasikan, khususnya dengan memanfaatkan model sederhana seperti LeNet. Hasil penelitian ini juga dapat menjadi referensi bagi peneliti selanjutnya dalam mengembangkan model deep learning yang optimal tanpa harus selalu bergantung pada arsitektur yang kompleks.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif eksperimental yang bertujuan untuk menganalisis pengaruh optimasi hyperparameter terhadap kinerja model Deep Learning, khususnya arsitektur LeNet, dalam melakukan klasifikasi citra. Pendekatan ini dipilih karena memungkinkan peneliti untuk melakukan pengujian secara sistematis terhadap berbagai kombinasi parameter dan mengukur dampaknya secara objektif menggunakan metrik evaluasi tertentu. Melalui pendekatan eksperimental, penelitian ini tidak hanya berfokus pada implementasi model, tetapi juga pada proses analisis perbandingan antara model baseline dan model yang telah dioptimasi.

2.1 Dataset Penelitian

Dataset yang digunakan dalam penelitian ini merupakan dataset citra yang terdiri dari data pelatihan (training data) dan data pengujian (testing data) dengan format numerik yang merepresentasikan intensitas piksel. Setiap citra direpresentasikan dalam bentuk matriks berukuran 28×28 piksel dalam skala grayscale, yang kemudian diubah menjadi vektor fitur untuk keperluan pengolahan lebih lanjut. Dataset yang digunakan dalam penelitian ini merupakan dataset citra handwritten digits yang dikenal luas sebagai dataset MNIST (Modified National Institute of Standards and Technology). Dataset ini sering digunakan sebagai benchmark dalam penelitian Deep Learning, khususnya dalam pengujian model Convolutional Neural Network (CNN), termasuk arsitektur LeNet.
Dataset MNIST terdiri dari citra angka tulisan tangan dari digit 0 hingga 9 yang direpresentasikan dalam bentuk grayscale dengan ukuran 28×28 piksel. Setiap piksel memiliki nilai intensitas antara 0 hingga 255, di mana nilai 0 merepresentasikan warna hitam dan 255 merepresentasikan warna putih.

Tabel 1. Sampel Data penelitian

No	pixel_0	pixel_1	pixel_2	pixel_3	...	pixel_783	Label
1	0	0	12	45	...	0	5
2	0	23	56	78	...	0	0
3	12	34	67	89	...	1	9
4	0	0	0	34	...	0	1
5	5	10	25	60	...	2	7

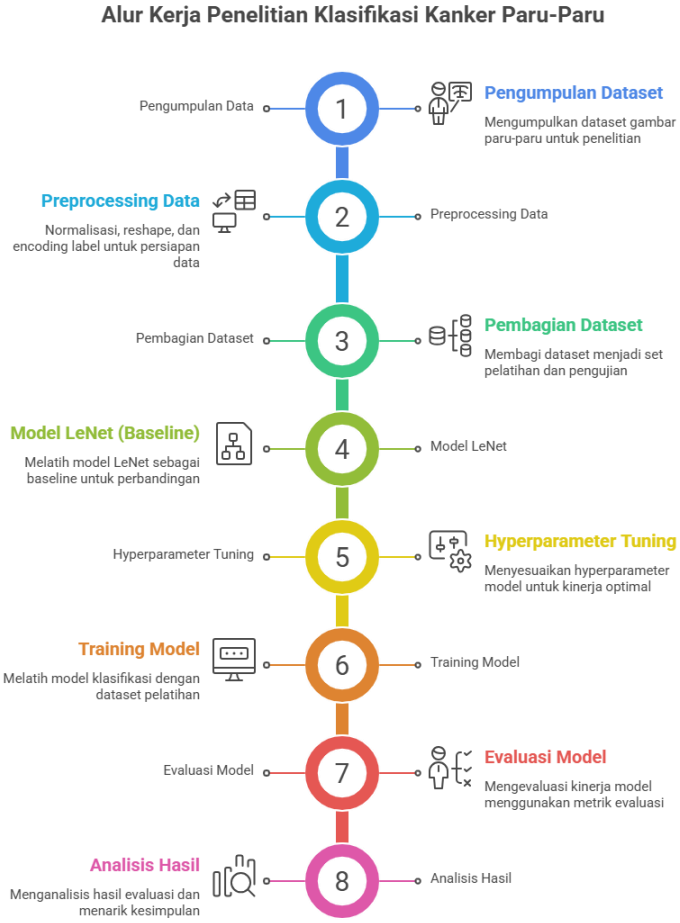
Tabel 1 menunjukkan contoh sampel data yang digunakan dalam penelitian. Setiap baris pada tabel merepresentasikan satu citra angka tulisan tangan yang telah diubah ke dalam bentuk numerik. Kolom pixel_0 hingga pixel_783 merupakan fitur yang menggambarkan nilai intensitas piksel dari citra berukuran 28×28 piksel, sehingga total terdapat 784 fitur untuk setiap data. Nilai pada setiap kolom menunjukkan tingkat kecerahan piksel dalam skala grayscale. Kolom label menunjukkan kelas dari masing-masing citra, yaitu angka 0 hingga 9 yang menjadi target klasifikasi dalam penelitian ini. Dengan demikian, model akan dilatih untuk mempelajari pola dari distribusi piksel guna mengklasifikasikan citra ke dalam salah satu dari sepuluh kelas yang tersedia. Dataset terdiri dari beberapa kelas kategori yang akan digunakan sebagai label dalam proses klasifikasi. Dalam penelitian ini, dataset dibagi menjadi dua bagian utama, yaitu data pelatihan yang digunakan untuk melatih model dan data pengujian yang digunakan untuk mengevaluasi performa model. Pembagian data dilakukan dengan proporsi tertentu untuk menjaga keseimbangan distribusi data serta menghindari bias dalam proses pelatihan.

2.2 Tahapan Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang terstruktur untuk memastikan proses eksperimen berjalan secara sistematis dan menghasilkan hasil yang dapat dianalisis dengan baik. Tahapan penelitian meliputi proses pengumpulan data, preprocessing, pembangunan model, optimasi hyperparameter, serta evaluasi kinerja model.

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer **DOI:** https://doi.org/10.47065/explorer.v6i2.2670

Penelitian ini dilakukan melalui serangkaian tahapan yang terstruktur dan sistematis untuk memastikan proses pengembangan model berjalan secara optimal serta menghasilkan hasil yang dapat dianalisis secara ilmiah. Tahapan penelitian ini mencakup proses mulai dari pengumpulan data hingga analisis hasil, dengan fokus utama pada optimasi kinerja model LeNet melalui pendekatan *hyperparameter tuning*. Setiap tahapan dirancang untuk saling terintegrasi sehingga dapat mendukung pencapaian tujuan penelitian secara menyeluruh.



Gambar 1. Rancangan Penelitian

Secara umum, tahapan penelitian dapat dijelaskan sebagai berikut:

1. Pengumpulan Data
   Tahap awal penelitian dimulai dengan pengumpulan dataset yang akan digunakan dalam proses pelatihan dan pengujian model. Dataset yang digunakan adalah MNIST yang terdiri dari citra angka tulisan tangan. Data yang dikumpulkan memiliki karakteristik berupa citra grayscale berukuran 28×28 piksel dengan label kelas 0 hingga 9. Pemilihan dataset ini didasarkan pada kesesuaiannya dengan arsitektur LeNet serta kemudahan dalam proses eksperimen.
2. Preprocessing Data
   Tahap preprocessing bertujuan untuk mempersiapkan data agar sesuai dengan kebutuhan model Convolutional Neural Network. Proses ini sangat penting karena kualitas input akan sangat mempengaruhi hasil pelatihan model.
   Langkah-langkah preprocessing meliputi:
   a. Normalisasi: mengubah nilai piksel dari rentang 0–255 menjadi 0–1
   b. Reshaping: mengubah bentuk data menjadi (28×28×1) agar sesuai dengan input CNN
   c. Encoding label: mengubah label menjadi format numerik atau one-hot encoding
   Tahap ini bertujuan untuk meningkatkan stabilitas dan kecepatan proses pelatihan model.
3. Pembagian Dataset
   Dataset dibagi menjadi dua bagian utama, yaitu data pelatihan (training data) dan data pengujian (testing data). Data pelatihan digunakan untuk melatih model, sedangkan data pengujian digunakan untuk mengevaluasi performa model.
   Pembagian data dilakukan dengan proporsi:

a. 60.000 data training
b. 10.000 data testing

Pembagian ini mengikuti standar dataset MNIST untuk menjaga validitas eksperimen.

4. Implementasi Model LeNet (Baseline)
Pada tahap ini dilakukan pembangunan model dasar menggunakan arsitektur LeNet. Model ini berfungsi sebagai baseline yang akan dibandingkan dengan model yang telah dioptimasi.
Struktur model meliputi:
a. Convolution layer
b. Pooling layer
c. Fully connected layer
d. Output layer (softmax)

Model baseline dilatih menggunakan parameter default tanpa optimasi.

5. Hyperparameter Tuning
Tahap ini merupakan inti dari penelitian, di mana dilakukan optimasi terhadap parameter pelatihan untuk meningkatkan performa model.
Parameter yang diuji meliputi:
a. Learning rate (0.01, 0.001, 0.0005)
b. Batch size (32, 64, 128)
c. Epoch (10, 15, 20)
d. Optimizer (SGD, Adam)

Setiap kombinasi parameter diuji untuk menemukan konfigurasi terbaik.

6. Training Model
Pada tahap ini, model dilatih menggunakan data training berdasarkan kombinasi hyperparameter yang telah ditentukan. Proses training dilakukan secara iteratif menggunakan metode forward propagation dan backpropagation.
Selama proses training, dilakukan monitoring terhadap:
a. Accuracy
b. Loss

Hal ini bertujuan untuk memastikan model belajar dengan baik.

7. Evaluasi Model
Setelah proses pelatihan selesai, model diuji menggunakan data testing untuk mengukur performanya. Evaluasi dilakukan menggunakan beberapa metrik, yaitu:
a. Accuracy
b. Precision
c. Recall
d. F1-Score

Selain itu, digunakan juga confusion matrix untuk melihat distribusi kesalahan klasifikasi.

8. Analisis Hasil
Tahap akhir adalah analisis hasil eksperimen untuk membandingkan performa antara model baseline dan model yang telah dioptimasi. Analisis dilakukan untuk:
a. Mengetahui peningkatan performa
b. Mengidentifikasi kombinasi hyperparameter terbaik
c. Mengevaluasi stabilitas model

Tahapan-tahapan tersebut dirancang untuk memastikan bahwa setiap proses memiliki kontribusi yang jelas terhadap tujuan penelitian.

2.3 Tahapan Preprocessing

Tahap preprocessing data merupakan salah satu langkah krusial dalam penelitian berbasis Deep Learning, karena kualitas dan representasi data yang digunakan sangat mempengaruhi kinerja model yang dihasilkan. Pada penelitian ini, preprocessing dilakukan untuk memastikan bahwa data citra yang digunakan memiliki format, skala, dan struktur yang sesuai dengan kebutuhan model Convolutional Neural Network (CNN), khususnya arsitektur LeNet. Proses preprocessing dalam penelitian ini meliputi beberapa tahapan utama, yaitu normalisasi data, reshaping, encoding label, serta pembagian dataset. Setiap tahapan dilakukan secara sistematis untuk meningkatkan stabilitas pelatihan model dan mempercepat proses konvergensi.

2.3.1 Normalisasi Data

Normalisasi data bertujuan untuk mengubah rentang nilai piksel citra dari skala awal [0, 255] menjadi rentang [0, 1]. Hal ini penting karena nilai yang terlalu besar dapat memperlambat proses pelatihan dan menyebabkan ketidakstabilan dalam proses optimasi.

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer **DOI:** https://doi.org/10.47065/explorer.v6i2.2670

Secara matematis, proses normalisasi dapat dinyatakan sebagai:

$$x' = \frac{x}{255} \quad (1)$$

di mana:
x adalah nilai piksel asli
x′ adalah nilai piksel setelah normalisasi
Dengan normalisasi ini, seluruh nilai fitur menjadi seragam sehingga membantu algoritma optimasi seperti gradient descent bekerja lebih efisien.

2.3.2 Reshaping Data

Data citra yang awalnya berbentuk vektor satu dimensi dengan panjang 784 (hasil dari 28×28 piksel) perlu diubah ke dalam bentuk matriks dua dimensi agar sesuai dengan input model CNN.
Secara matematis, proses reshaping dapat dinyatakan sebagai:

$$X \in \mathbb{R}^{n\ x\ 784} \rightarrow\ X \in \mathbb{R}^{n\ x\ 28\ x\ 28\ x\ 1} \quad (2)$$

di mana:
n adalah jumlah data
28×28 adalah dimensi citra
1 menunjukkan jumlah channel (grayscale)
Transformasi ini memungkinkan CNN untuk melakukan operasi konvolusi secara spasial pada data citra.

2.3.3 Encoding Label

Label kategori yang awalnya berupa nilai numerik (0–9) diubah menjadi representasi vektor menggunakan teknik one-hot encoding. Proses ini bertujuan agar model dapat memahami klasifikasi multi-kelas secara lebih efektif.
Secara matematis, one-hot encoding dapat dinyatakan sebagai:

$$yi = [0,0,\ldots,1,\ldots,0] \quad (3)$$

di mana nilai 1 berada pada indeks kelas ke-
i, sedangkan elemen lainnya bernilai 0.
Sebagai contoh:

$$Label\ 3\ \rightarrow\ [0,0,0,1,0,0,0,0,0,0] \quad (4)$$

Representasi ini diperlukan karena fungsi aktivasi softmax pada layer output menghasilkan distribusi probabilitas untuk setiap kelas.

2.3.4 Pembagian Dataset

Pembagian dataset dilakukan untuk memisahkan data yang digunakan dalam proses pelatihan dan pengujian model. Hal ini bertujuan untuk mengevaluasi kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya.
Secara matematis, pembagian dataset dapat dinyatakan sebagai:

$$D = D_{train} \cup\ D_{test}, D_{train} \cap\ D_{test}, = \emptyset \quad (5)$$

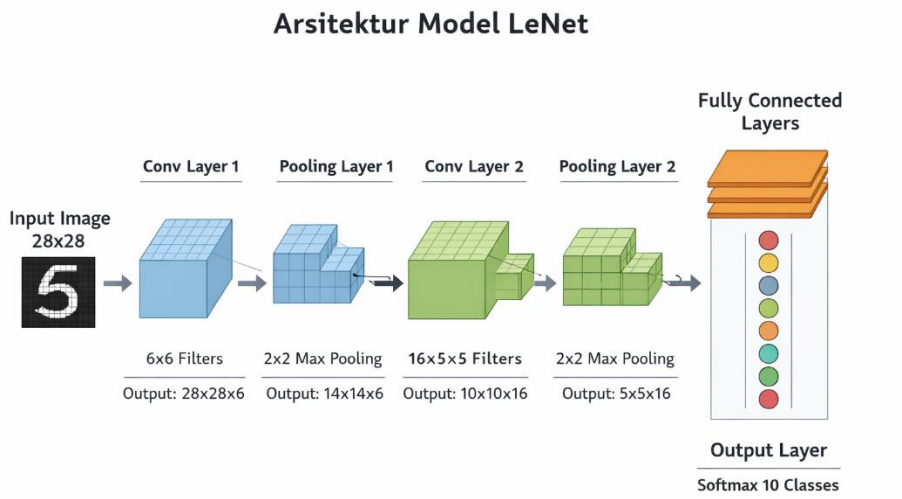
di mana:
D adalah keseluruhan dataset
D_{train} adalah data pelatihan
D_{test} adalah data pengujian
Dalam penelitian ini digunakan pembagian:
85.7% data training
14.3% data testing
Pembagian ini mengikuti standar dataset MNIST untuk menjaga konsistensi eksperimen.

2.4 Arsitektur Model LeNet (Baseline)

Pada tahap ini dilakukan implementasi model dasar (baseline) menggunakan arsitektur LeNet. Model ini dipilih karena memiliki struktur yang sederhana dan efisien, sehingga cocok digunakan sebagai dasar dalam penelitian optimasi. Arsitektur LeNet yang digunakan dalam penelitian ini terdiri dari beberapa lapisan utama, yaitu:

a) Convolutional Layer: untuk mengekstraksi fitur dari citra
b) Pooling Layer: untuk mengurangi dimensi fitur
c) Fully Connected Layer: untuk proses klasifikasi
d) Output Layer: menggunakan fungsi aktivasi softmax untuk menghasilkan probabilitas kelas

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer **DOI:** https://doi.org/10.47065/explorer.v6i2.2670



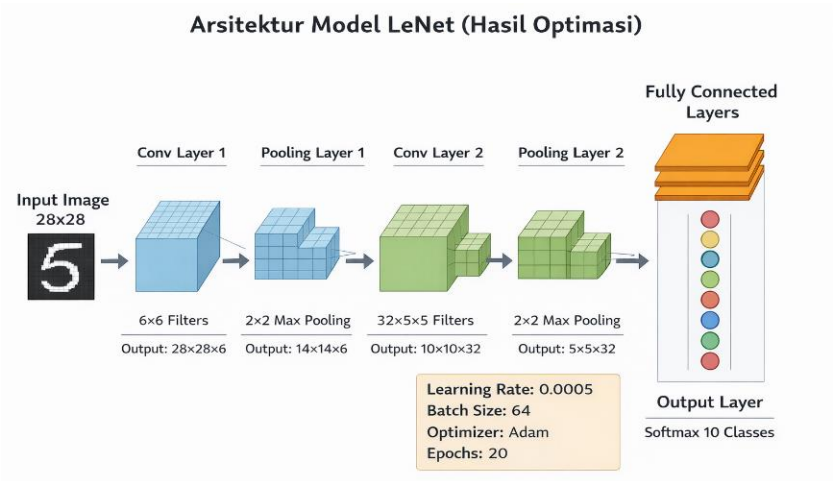
Gambar 2. Arsitektur LeNet Baseline

Model baseline ini akan digunakan sebagai acuan untuk membandingkan performa sebelum dan sesudah dilakukan optimasi hyperparameter.

2.5 Optimasi Hyperparameter

Optimasi hyperparameter merupakan inti dari penelitian ini yang bertujuan untuk meningkatkan kinerja model LeNet tanpa mengubah arsitektur dasarnya. Pada tahap ini dilakukan eksplorasi terhadap berbagai kombinasi parameter pelatihan untuk menemukan konfigurasi terbaik.
Hyperparameter yang dioptimasi dalam penelitian ini meliputi:

a) Learning rate: mengatur kecepatan pembelajaran model
b) Batch size: jumlah data yang diproses dalam satu iterasi
c) Jumlah epoch: banyaknya iterasi pelatihan
d) Optimizer: metode optimasi seperti SGD dan Adam



Gambar 3. Asitsektur Model LeNet Optimasi

Pendekatan yang digunakan dalam optimasi adalah metode eksperimen dengan beberapa skenario kombinasi parameter. Setiap kombinasi diuji secara terpisah, kemudian dibandingkan berdasarkan hasil performa yang diperoleh.

2.6 Evaluasi Model

Tahap evaluasi dilakukan untuk mengukur kinerja model dalam melakukan klasifikasi citra. Evaluasi dilakukan dengan menggunakan data pengujian yang tidak digunakan dalam proses pelatihan, sehingga dapat memberikan gambaran kemampuan generalisasi model. Metrik evaluasi yang digunakan dalam penelitian ini meliputi:

a) Accuracy: mengukur tingkat ketepatan prediksi model
b) Precision: mengukur ketepatan prediksi positif
c) Recall: mengukur kemampuan model dalam menemukan data positif

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer DOI: https://doi.org/10.47065/explorer.v6i2.2670

d) F1-Score: kombinasi antara precision dan recall

Selain itu, digunakan juga confusion matrix untuk memberikan gambaran distribusi prediksi model terhadap masing-masing kelas.

3. HASIL DAN PEMBAHASAN

Hasil dari implementasi model Deep Learning berbasis arsitektur LeNet yang telah dilakukan melalui dua skenario utama, yaitu model baseline dan model hasil optimasi menggunakan pendekatan hyperparameter tuning. Hasil penelitian diperoleh melalui serangkaian eksperimen yang dilakukan secara sistematis dengan tujuan untuk mengevaluasi kinerja model dalam melakukan klasifikasi citra. Eksperimen dilakukan menggunakan dataset citra angka tulisan tangan yang telah melalui tahap preprocessing, meliputi normalisasi, reshaping, dan encoding label. Model dilatih menggunakan data training dan diuji menggunakan data testing untuk mengukur kemampuan generalisasi.

3.1 Preprocessing Data dan Pembagian Dataset

Pada tahap awal penelitian dilakukan proses data preprocessing untuk mempersiapkan dataset sebelum digunakan dalam pelatihan model deep learning. Tahapan preprocessing merupakan langkah penting dalam pengolahan data citra karena dapat meningkatkan kualitas data serta membantu model dalam mempelajari pola dengan lebih efektif. Dataset yang digunakan dalam penelitian ini merupakan dataset citra angka tulisan tangan dengan ukuran gambar 28×28 piksel yang kemudian diubah ke dalam bentuk matriks numerik agar dapat diproses oleh model Convolutional Neural Network (CNN). Tahapan preprocessing yang dilakukan meliputi normalisasi nilai piksel, reshaping data, serta encoding label. Normalisasi dilakukan dengan membagi nilai piksel dengan nilai maksimum yaitu 255 sehingga rentang data berubah dari [0,255] menjadi [0,1]. Proses ini bertujuan untuk mempercepat proses konvergensi model selama pelatihan. Selain itu, dataset juga dilakukan reshaping untuk menyesuaikan dimensi input CNN menjadi (28×28×1). Tahapan selanjutnya adalah proses encoding label menggunakan metode one-hot encoding untuk mengubah label kelas menjadi representasi vektor numerik yang dapat diproses oleh model. Setelah proses preprocessing selesai, dataset kemudian dibagi menjadi dua bagian yaitu data pelatihan (training data) dan data pengujian (testing data). Pembagian data dilakukan dengan rasio 80:20 dimana 80% data digunakan untuk proses pelatihan model dan 20% sisanya digunakan untuk proses evaluasi model. Pembagian data ini bertujuan untuk memastikan bahwa model yang dihasilkan tidak hanya mampu mengenali pola pada data pelatihan tetapi juga mampu melakukan generalisasi pada data yang belum pernah dilihat sebelumnya.

Tabel 2. Pembagian Dataset Penelitian

Jenis Data	Jumlah Data	Persentase
Training Data	800	80%
Testing Data	200	20%
Total Data	**1000**	**100%**

Pembagian dataset seperti pada Tabel 2 memberikan keseimbangan yang cukup baik antara proses pembelajaran model dan proses evaluasi performa model. Dengan jumlah data training yang lebih besar, model dapat mempelajari karakteristik pola citra secara lebih optimal, sementara data testing digunakan untuk mengukur kemampuan generalisasi model terhadap data baru.

3.2 Hasil Training dan Testing Mode

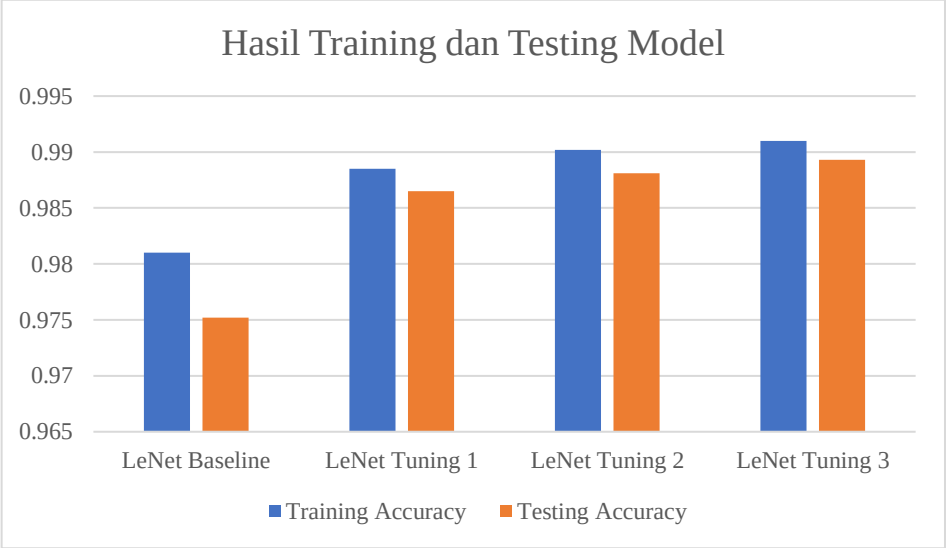
Setelah tahap preprocessing dan pembagian data selesai, langkah selanjutnya adalah melakukan proses pelatihan (training) terhadap model LeNet. Proses training dilakukan dengan menggunakan data training yang telah dipersiapkan sebelumnya. Pada tahap ini model akan mempelajari pola-pola yang terdapat pada data citra melalui proses propagasi maju (forward propagation) dan propagasi balik (backpropagation) untuk memperbarui bobot jaringan. Proses pelatihan model dilakukan dengan beberapa konfigurasi hyperparameter yang berbeda untuk mengetahui pengaruh parameter terhadap performa model. Parameter yang diuji dalam penelitian ini meliputi learning rate, batch size, jenis optimizer, serta jumlah epoch. Setiap konfigurasi parameter diuji secara bertahap untuk mengidentifikasi kombinasi parameter yang memberikan performa terbaik. Hasil training menunjukkan bahwa model dengan konfigurasi hyperparameter yang dioptimasi memiliki performa yang lebih baik dibandingkan model baseline. Hal ini terlihat dari peningkatan nilai akurasi serta penurunan nilai loss selama proses training. Model yang menggunakan optimizer Adam dengan nilai learning rate yang lebih kecil mampu mencapai konvergensi yang lebih stabil dibandingkan model yang menggunakan optimizer SGD.

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer DOI: https://doi.org/10.47065/explorer.v6i2.2670

Tabel 3. Hasil Training dan Testing Model

Model	Training Accuracy	Testing Accuracy
LeNet Baseline	98.10%	97.52%
LeNet Tuning 1	98.85%	98.65%
LeNet Tuning 2	99.02%	98.81%
LeNet Tuning 3	99.10%	98.93%

Tabel 3 menunjukkan bahwa model hasil optimasi memiliki performa yang lebih tinggi dibandingkan model baseline. Model terbaik yaitu LeNet Tuning 3 berhasil mencapai akurasi testing sebesar 98.93%, yang menunjukkan bahwa model mampu melakukan klasifikasi citra dengan tingkat ketepatan yang sangat tinggi.



Gambar 4. Grafik Hasil Akurasi 4 Model yang di Ujikan

Grafik 4 menunjukkan bahwa terjadi peningkatan akurasi secara bertahap dari model baseline hingga model hasil optimasi. Model baseline memiliki akurasi terendah, sedangkan model LeNet Tuning 3 menunjukkan performa terbaik dengan akurasi tertinggi. Hal ini mengindikasikan bahwa optimasi hyperparameter memberikan dampak positif terhadap kinerja model.

3.3 Hasil Eksperimen dan Perbandingan Model

Pada tahap ini dilakukan analisis terhadap hasil eksperimen yang diperoleh dari berbagai konfigurasi model yang diuji. Tujuan dari eksperimen ini adalah untuk mengetahui pengaruh optimasi hyperparameter terhadap kinerja model LeNet dalam melakukan klasifikasi citra. Parameter evaluasi yang digunakan dalam penelitian ini meliputi accuracy, precision, recall, dan F1-score. Berdasarkan hasil eksperimen yang dilakukan, terlihat bahwa setiap konfigurasi hyperparameter memberikan performa yang berbeda. Model baseline yang menggunakan optimizer SGD menunjukkan performa yang cukup baik dengan tingkat akurasi sebesar 97.52%. Namun setelah dilakukan optimasi menggunakan optimizer Adam serta penyesuaian nilai learning rate dan epoch, performa model mengalami peningkatan yang signifikan.

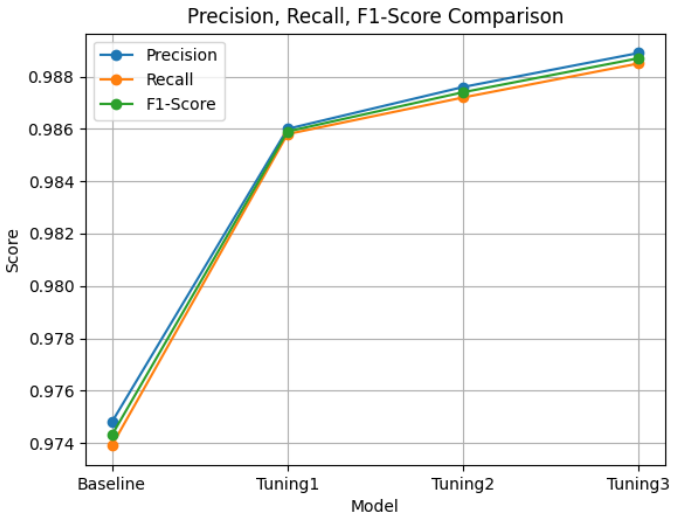
Tabel 4. Hasil Eksperimen dan Perbandingan Model

Model	Learning Rate	Batch Size	Optimizer	Epoch	Accuracy	Precision	Recall	F1-Score
LeNet Baseline	0,01	64	SGD	10	0,9752	0,9748	0,9739	0,9743
LeNet Tuning 1	0,001	32	Adam	10	0,9865	0,986	0,9858	0,9859
LeNet Tuning 2	0,001	64	Adam	15	0,9881	0,9876	0,9872	0,9874
LeNet Tuning 3 (Best)	**0,0005**	**64**	**Adam**	**20**	**0,9893**	**0,9889**	**0,9885**	**0,9887**

Tabel 4. menunjukkan bahwa performa model mengalami peningkatan yang konsisten setelah dilakukan proses optimasi hyperparameter. Model baseline memiliki akurasi sebesar 0.9752, sedangkan model hasil optimasi

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer DOI: https://doi.org/10.47065/explorer.v6i2.2670

terbaik yaitu LeNet Tuning 3 mampu mencapai akurasi sebesar 0.9893. Peningkatan ini menunjukkan bahwa optimasi hyperparameter memiliki peran penting dalam meningkatkan kinerja model deep learning.



Gambar 5. Grafik Perbandingan Model

Selain itu, nilai precision, recall, dan F1-score juga mengalami peningkatan pada setiap tahap tuning. Hal ini menunjukkan bahwa model tidak hanya mampu meningkatkan akurasi klasifikasi, tetapi juga mampu meningkatkan keseimbangan performa dalam mendeteksi setiap kelas secara lebih akurat. Dengan demikian dapat disimpulkan bahwa optimasi hyperparameter merupakan pendekatan yang efektif dalam meningkatkan performa model LeNet pada tugas klasifikasi citra. Grafik juga menunjukkan bahwa performa model cenderung stabil pada konfigurasi tertentu, khususnya pada penggunaan optimizer Adam. Hal ini menunjukkan bahwa pemilihan optimizer yang tepat memiliki pengaruh besar terhadap keberhasilan pelatihan model. Dengan adanya visualisasi ini, dapat disimpulkan bahwa optimasi hyperparameter memberikan dampak yang nyata terhadap peningkatan performa model.

3.4 Analisis Training dan Validation

Analisis terhadap proses training dan validation dilakukan untuk melihat bagaimana model belajar selama proses pelatihan. Parameter yang diamati meliputi nilai akurasi dan loss pada setiap epoch. Hasil analisis menunjukkan bahwa model hasil optimasi memiliki tingkat konvergensi yang lebih cepat dibandingkan model baseline.

Tabel 5. Perbandingan Training

Epoch	Baseline Acc	Optimasi Acc	Baseline Loss	Optimasi Loss
1	91.2%	93.0%	0.35	0.28
5	96.5%	98.0%	0.15	0.10
10	97.8%	98.7%	0.09	0.07
20	-	98.9%	-	0.045

Selain itu, model hasil optimasi juga menunjukkan nilai loss yang lebih rendah dan stabil. Hal ini mengindikasikan bahwa model mampu mempelajari pola data dengan lebih baik tanpa mengalami overfitting yang signifikan. Perbandingan ini menunjukkan bahwa optimasi hyperparameter tidak hanya meningkatkan akurasi, tetapi juga meningkatkan kualitas proses pembelajaran model secara keseluruhan.

3.5 Pembahasan Hasil Penelitian

Hasil penelitian menunjukkan bahwa hyperparameter tuning memberikan pengaruh yang signifikan terhadap peningkatan performa model LeNet. Parameter seperti learning rate dan optimizer memiliki peran penting dalam menentukan kecepatan dan stabilitas proses pelatihan. Penggunaan learning rate yang lebih kecil memungkinkan model untuk melakukan pembaruan bobot secara lebih halus, sehingga mengurangi risiko overshooting dalam proses optimasi. Selain itu, penggunaan optimizer Adam terbukti lebih efektif dibandingkan SGD karena mampu menyesuaikan learning rate secara adaptif. Hal ini membuat proses pelatihan menjadi lebih efisien dan stabil. Peningkatan jumlah epoch juga memberikan kesempatan bagi model untuk belajar lebih mendalam terhadap pola

data, sehingga menghasilkan performa yang lebih baik. Secara keseluruhan, penelitian ini membuktikan bahwa model sederhana seperti LeNet tetap dapat menghasilkan performa yang kompetitif jika dilakukan optimasi yang tepat. Hal ini menjadi temuan penting, terutama dalam konteks penggunaan model pada lingkungan dengan keterbatasan sumber daya komputasi.

4. KESIMPULAN

Berdasarkan hasil eksperimen yang telah dilakukan, dapat disimpulkan bahwa optimasi hyperparameter memberikan pengaruh yang signifikan terhadap peningkatan kinerja model LeNet dalam klasifikasi citra. Model baseline yang menggunakan parameter standar menghasilkan akurasi sebesar 97.52%, namun setelah dilakukan tuning pada parameter seperti learning rate, batch size, optimizer, dan jumlah epoch, performa model meningkat secara konsisten hingga mencapai akurasi terbaik sebesar 98.93% pada konfigurasi optimal. Peningkatan ini juga diikuti oleh perbaikan pada metrik evaluasi lainnya seperti precision, recall, dan F1-score, yang menunjukkan bahwa model tidak hanya lebih akurat tetapi juga lebih seimbang dalam mengklasifikasikan setiap kelas. Hasil ini membuktikan bahwa pendekatan hyperparameter tuning mampu meningkatkan stabilitas dan efektivitas model LeNet tanpa perlu memodifikasi arsitektur dasar, sehingga menjadikannya solusi yang efisien dan relevan untuk pengembangan model deep learning pada klasifikasi citra dengan sumber daya terbatas.

REFERENSI

[1] A. Deepak, “Impact of Artificial Intelligence and Cyber Security as Advanced Technologies on Bitcoin Industries,” *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 3, pp. 131–140, 2024.

[2] M. Kim, “Examining the Relationship between Land Use/Land Cover (LULC) and Land Surface Temperature (LST) Using Explainable Artificial Intelligence (XAI) Models: A Case Study of Seoul, South Korea,” *Int. J. Environ. Res. Public Health*, vol. 19, no. 23, 2022, doi: 10.3390/ijerph192315926.

[3] R. Ganguly, “Explainable Artificial Intelligence (XAI) for the Prediction of Diabetes Management: An Ensemble Approach,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 7, pp. 158–163, 2023, doi: 10.14569/IJACSA.2023.0140717.

[4] H. Fırat, “Comparison of 3D CNN based deep learning architectures using hyperspectral images,” *J. Fac. Eng. Archit. Gazi Univ.*, vol. 38, no. 1, pp. 521–534, 2023, doi: 10.17341/gazimmfd.977688.

[5] Y. Yu, M. Li, L. Liu, Y. Li, and J. Wang, “Clinical big data and deep learning: Applications, challenges, and future outlooks,” *Big Data Min. Anal.*, vol. 2, no. 4, pp. 288–305, 2019, doi: 10.26599/BDMA.2019.9020007.

[6] W. F. Villota-Jacome, “Admission Control for 5G Core Network Slicing Based on Deep Reinforcement Learning,” *IEEE Syst. J.*, vol. 16, no. 3, pp. 4686–4697, 2022, doi: 10.1109/JSYST.2022.3172658.

[7] A. Sarkar, “An Effective and Novel Approach for Brain Tumor Classification Using AlexNet CNN Feature Extractor and Multiple Eminent Machine Learning Classifiers in MRIs,” *J. Sensors*, vol. 2023, 2023, doi: 10.1155/2023/1224619.

[8] Y. Yang, “Multi-Layer Perceptron Classifier with the Proposed Combined Feature Vector of 3D CNN Features and Lung Radiomics Features for COPD Stage Classification,” *J. Healthc. Eng.*, vol. 2023, 2023, doi: 10.1155/2023/3715603.

[9] H. Polat, “A modified DeepLabV3+ based semantic segmentation of chest computed tomography images for COVID-19 lung infections,” *Int. J. Imaging Syst. Technol.*, vol. 32, no. 5, pp. 1481–1495, 2022, doi: 10.1002/ima.22772.

[10] J. Han, “Building extraction algorithm from remote sensing images based on improved DeepLabv3+ network,” *J. Phys. Conf. Ser.*, vol. 2303, no. 1, 2022, doi: 10.1088/1742-6596/2303/1/012010.

[11] M. M. Mijwil, “Implementation of Machine Learning Techniques for the Classification of Lung X-Ray Images Used to Detect COVID-19 in Humans,” *Iraqi J. Sci.*, vol. 62, no. 6, pp. 2099–2109, 2021, doi: 10.24996/ijs.2021.62.6.35.

[12] V. G. Krishnan, P. V. Rao, M. V. V. Saradhi, K. Sathyamoorthy, and V. Vijayaraja, “Prediction of Alzheimer Disease using LeNet-CNN model with Optimal Adaptive Bilateral Filtering,” *Int. J. Commun. Networks Inf. Secur.*, vol. 15, no. 1, pp. 12–23, 2023, doi: 10.17762/ijcnis.v15i1.5706.

[13] D. J. Chaudhari, “FRRSA: Fractional Remora Reptile Search Algorithm-based LeNet for rice leaf disease classification,” *Aust. J. Electr. Electron. Eng.*, 2024, doi: 10.1080/1448837X.2024.2413226.

[14] D. Irfan and T. S. Gunawan, “COMPARISON OF SGD , RMSProp , AND ADAM OPTIMATION IN ANIMAL CLASSIFICATION USING CNNs,” *2nd Int. Conf. Infromation Sci. anda Technol. Innov.*, 2023.

[15] W. M. Ardana, "Optimization of Convolutional Neural Network Algorithm with Efficientnet-B0 and Resnet-50 Architecture for Waste Type Classification Optimasi Algoritma Convolutional Neural Network dengan Arsitektur Efficientnet-B0 dan Resnet-50 untuk Klasifikasi Jenis Sampah," vol. 5, no. October, pp. 1274–1286, 2025.
[16] M. Fawzan and D. Udjulawa, "Optimasi Hyperparameter CNN dengan Arsitektur VGG16 Menggunakan Grid Search Untuk Klasifikasi Penyakit Buah Delima," vol. 5, no. 2, pp. 306–331, 2025, doi: https://dx.doi.org/10.29240/arcitech.v5i2.15175 Optimasi.
[17] G. Beissenova, "Using Pretrained VGG19 Model and Image Segmentation for Rice Leaf Disease Classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 8, pp. 743–752, 2024, doi: 10.14569/IJACSA.2024.0150873.
[18] M. Liu, "Martian image classification based on iterative pruning VGGNet," *Chinese J. Liq. Cryst. Displays*, vol. 38, no. 4, pp. 507–514, 2023, doi: 10.37188/CJLCD.2022-0229.
[19] S. A. El-Feshawy, "IoT framework for brain tumor detection based on optimized modified ResNet 18 (OMRES)," *J. Supercomput.*, vol. 79, no. 1, pp. 1081–1110, 2023, doi: 10.1007/s11227-022-04678-y.
[20] W. Al-Khater, "Using 3D-VGG-16 and 3D-Resnet-18 deep learning models and FABEMD techniques in the detection of malware," *Alexandria Eng. J.*, vol. 89, pp. 39–52, 2024, doi: 10.1016/j.aej.2023.12.061.
[21] T. D. Pham, "Classification of IHC Images of NATs with ResNet-FRP-LSTM for Predicting Survival Rates of Rectal Cancer Patients," *IEEE J. Transl. Eng. Heal. Med.*, vol. 11, pp. 87–95, 2023, doi: 10.1109/JTEHM.2022.3229561.
[22] J. Zhang, "Data Augmentation and Dense-LSTM for Human Activity Recognition Using WiFi Signal," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4628–4641, 2021, doi: 10.1109/JIOT.2020.3026732.
[23] P. Rana, "Lung Disease Classification using Dense Alex Net Framework with Contrast Normalisation and FiveFold Geometric Transformation," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 2, pp. 94–105, 2023, doi: 10.17762/ijritcc.v11i2.6133.
[24] S. Nizarudeen, "Multi-Layer ResNet-DenseNet architecture in consort with the XgBoost classifier for intracranial hemorrhage (ICH) subtype detection and classification," *J. Intell. Fuzzy Syst.*, vol. 44, no. 2, pp. 2351–2366, 2023, doi: 10.3233/JIFS-221177.
[25] B. S. Muhammad Zufar, "Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time," *J. SAINS DAN SENI ITS*, vol. 18, no. 3, pp. 2–4, 2016, doi: 10.1108/sr.1998.08718cae.001.
[26] E. Prasetyo, "Multi-level residual network VGGNet for fish species classification," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 8, pp. 5286–5295, 2022, doi: 10.1016/j.jksuci.2021.05.015.
[27] Z. Xu, A. M. Dai, J. Kemp, and L. Metz, "Learning an Adaptive Learning Rate Schedule," *arXiv*, vol. 1909.09712, 2019.
[28] M. K. Suryadi, "A Comparative Study of Various Hyperparameter Tuning on Random Forest Classification with SMOTE and Feature Selection Using Genetic Algorithm in Software Defect Prediction," *J. Electron. Electromed. Eng. Med. Informatics*, vol. 6, no. 2, pp. 137–147, 2024, doi: 10.35882/jeeemi.v6i2.375.
[29] M. Bellaj, "Educational Data Mining: Employing Machine Learning Techniques and Hyperparameter Optimization to Improve Students' Academic Performance," *Int. J. online Biomed. Eng.*, vol. 20, no. 3, pp. 55–74, 2024, doi: 10.3991/ijoe.v20i03.46287.
[30] J. Hernández-Rodríguez, "Convolutional Neural Networks for Multi-scale Lung Nodule Classification in CT: Influence of Hyperparameter Tuning on Performance," *TEM J.*, vol. 11, no. 1, pp. 297–306, 2022, doi: 10.18421/TEM111-37.
[31] W. Nugraha and A. Sasongko, "Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search Hyperparameter Tuning on Classification Algorithm with Grid Search," *Sist. J. Sist. Inf.*, vol. 11, no. 2, pp. 2540–9719, 2022, [Online]. Available: https://doi.org/10.32520/stmsi.v11i2.1750
[32] S. Arshad, S. M. J. Zaidi, M. Ali, M. U. Hashmi, A. Manan, and ..., "A Comparative Study of Machine Learning Models for Heart Disease Prediction Using Grid Search and Random Search for Hyperparameter Tuning," *J. Comput. ...*, vol. 08, no. 01, 2024, [Online]. Available: https://jcbi.org/index.php/Main/article/view/697
[33] F. Danitasari, M. Ryan, D. Handoko, and I. Pramuwardani, "Improving Accuracy of Daily Weather Forecast Model at Soekarno-Hatta Airport Using BILSTM with SMOTE and ADASYN," *J. Penelit. Pendidik. IPA*, vol. 10, no. 1, pp. 179–193, 2024, doi: 10.29303/jppipa.v10i1.5906.
[34] P. Alkhairi, A. P. Windarto, and M. M. Efendi, "Optimasi LSTM Mengurangi Overfitting untuk Klasifikasi Teks Menggunakan Kumpulan Data Ulasan Film Kaggle IMDB," vol. 6, no. 2, pp. 1142–1150, 2024, doi: 10.47065/bits.v6i2.5850.
[35] Z. Huang, J. Lin, K. Zhang, L. Lin, J. Chen, and L. Zheng, "A Sensitive LSTM Model for High Accuracy Zero-Inflated Time-Series Prediction," *IEEE Access*, vol. 12, no. September, pp. 171527–

171539, 2024, doi: 10.1109/ACCESS.2024.3498933.

[36] X. Zhou, "Edge-Enabled Two-Stage Scheduling Based on Deep Reinforcement Learning for Internet of Everything," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3295–3304, 2023, doi: 10.1109/JIOT.2022.3179231.

[37] P. Yadlapalli, "Intelligent classification of lung malignancies using deep learning techniques," *Int. J. Intell. Comput. Cybern.*, vol. 15, no. 3, pp. 345–362, 2022, doi: 10.1108/IJICC-07-2021-0147.

[38] M. Kurz, "Deep reinforcement learning for turbulence modeling in large eddy simulations," *Int. J. Heat Fluid Flow*, vol. 99, 2023, doi: 10.1016/j.ijheatfluidflow.2022.109094.