

Optimasi Kinerja Arsitektur CNN Ringan Menggunakan Pendekatan Bayesian untuk Identifikasi Skrip Bima

Performance Optimization of Lightweight CNN Architecture Using Bayesian Approach for Bima Script Identification

Dayang Aisyah^{a,1}, Muhammad Faisal^{a,2,*}, Lukman Anas^{a,3}, Abd Rakhim Nanda^{b,4}, Syadiah Nor Wan Shamsuddin^{c,5} dan Muhammad Syafaat S. Kuba^{b,6}

^a Informatika, Universitas Muhammadiyah Makassar, Makassar, Indonesia

^b Teknik Pengairan, Universitas Muhammadiyah Makassar, Makassar, Indonesia

^c Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia

¹105841109822@student.unismuh.ac.id; ²muhfaisal@unismuh.ac.id; ³lukmananas@unismuh.ac.id;

⁴abd.rakhimnanda@unismuh.ac.id; ⁵syadiah@unisza.edu.my; ⁶syafaat_skuba@unismuh.ac.id

*corresponding author

Informasi Artikel

Diserahkan : 29 April 2026
Diterima : 16 Mei 2026
Direvisi : 18 Mei 2026
Diterbitkan : 21 Mei 2026

Kata Kunci:

Aksara Bima
Lightweight CNN
Bayesian Optimization
Klasifikasi Citra
Pelestarian Digital

Keywords:

Bima Script
Lightweight CNN
Bayesian Optimization
Image Classification
Digital Preservation

This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



ABSTRAK

Identifikasi aksara daerah penting untuk mendukung pelestarian budaya digital, namun masih terkendala keterbatasan dataset, kemiripan karakter, dan kebutuhan model yang efisien. Penelitian ini mengoptimasi arsitektur Lightweight CNN menggunakan Bayesian Optimization untuk identifikasi aksara Bima. Dataset terdiri atas 6.190 citra aksara Bima dalam 44 kelas, mencakup aksara Bima baru dan lama. Model menggunakan MobileNetV3-Large sebagai backbone dengan optimasi learning rate, dropout, batch size, dan konfigurasi fine-tuning melalui Tree-structured Parzen Estimator. Hasil eksperimen menunjukkan accuracy 93,06%, precision 92,26%, recall 92,55%, dan F1-score 91,91%, lebih unggul dibanding machine learning tradisional, CNN konvensional, dan beberapa CNN ringan modern. Target accuracy 90% dicapai pada trial keempat. Dengan 3.253.676 parameter dan waktu inferensi 63,35 ms per citra, model ini terbukti akurat, efisien, dan berpotensi diterapkan pada digitalisasi manuskrip serta OCR aksara daerah.

ABSTRACT

Regional script identification is important to support the preservation of digital culture, but it is still constrained by dataset limitations, character similarities, and the need for efficient models. This study optimizes the Lightweight CNN architecture using Bayesian Optimization for Bima script identification. The dataset consists of 6,190 images of the Bima script in 44 classes, covering the new and old Bima scripts. The model uses MobileNetV3-Large as the backbone with learning rate optimization, dropout, batch size, and fine-tuning configuration via a Tree-structured Parzen Estimator. The results of the experiment showed an accuracy of 93.06%, precision of 92.26%, recall of 92.55%, and an F1-score of 91.91%, superior to traditional machine learning, conventional CNNs, and some modern lightweight CNNs. The target accuracy of 90% was achieved in the fourth trial. With 3,253,676 parameters and an inference time of 63.35 ms per image, this model has proven to be accurate, efficient, and potentially applicable to manuscript digitization and regional script OCR.

I. Pendahuluan

Perkembangan kecerdasan buatan, khususnya *deep learning*, telah mendorong kemajuan signifikan pada bidang klasifikasi citra, deteksi objek, pengenalan pola, dan analisis visual otomatis. Di antara berbagai metode yang berkembang, *Convolutional Neural Network* (CNN) menjadi pendekatan utama karena mampu mengekstraksi fitur citra secara otomatis serta memberikan performa tinggi pada data kompleks. CNN telah banyak diterapkan pada sistem transportasi cerdas, inspeksi industri, pengenalan karakter, dan berbagai aplikasi klasifikasi modern. Namun, arsitektur CNN konvensional umumnya memiliki jumlah parameter besar dan kebutuhan komputasi tinggi sehingga kurang efisien untuk perangkat dengan sumber daya terbatas [1], [2].

Indonesia memiliki kekayaan budaya lokal yang sangat luas, termasuk sistem tulisan tradisional daerah. Salah satu warisan budaya tersebut adalah aksara Bima atau aksara Mbojo yang berasal dari wilayah Bima,

Nusa Tenggara Barat. Aksara ini telah digunakan sejak masa kerajaan Bima sebagai media penulisan naskah, administrasi, dan komunikasi masyarakat [3]. Dalam kajian Islam, pelestarian ilmu pengetahuan dan warisan sejarah dipandang sebagai bagian penting dalam menjaga keberlangsungan peradaban dan transfer nilai antargenerasi. Tradisi intelektual Islam juga menempatkan pengembangan ilmu pengetahuan, dokumentasi, dan pemeliharaan khazanah budaya sebagai unsur yang memiliki nilai kemanfaatan bagi masyarakat [4]. Namun demikian, eksistensi aksara Bima saat ini semakin mengalami penurunan yang dipengaruhi oleh terbatasnya dokumentasi berbasis digital, kurangnya media pembelajaran yang adaptif, serta rendahnya penggunaan di kalangan generasi muda [5]. Oleh sebab itu, pengembangan digitalisasi dan sistem identifikasi otomatis menjadi salah satu upaya strategis untuk mendukung pelestarian aksara daerah secara berkelanjutan.

Berbagai penelitian sebelumnya menunjukkan bahwa CNN efektif digunakan untuk pengenalan tulisan tangan dan klasifikasi karakter. Pada penelitian aksara Bima terdahulu, CNN dengan *dataset* 2.640 citra dan 22 kelas mampu mencapai akurasi 97,34%. Pada penelitian lain, CNN juga berhasil digunakan untuk pengenalan karakter tradisional dan klasifikasi visual multi-kelas dengan tingkat akurasi tinggi. Hasil tersebut menunjukkan bahwa CNN memiliki kemampuan kuat dalam mempelajari pola bentuk karakter yang kompleks dan bervariasi [6], [7]. Meskipun demikian, masih terdapat beberapa keterbatasan pada penelitian sebelumnya. Sebagian besar studi pada aksara daerah masih menggunakan CNN standar dengan jumlah parameter relatif besar dan belum menekankan efisiensi model. Selain itu, jumlah kelas yang digunakan masih terbatas dibanding variasi karakter sebenarnya. Penentuan hyperparameter seperti *learning rate*, *batch size*, *dropout*, jumlah filter, dan kedalaman *layer* juga umumnya dilakukan secara manual atau *trial and error* [8]. Pendekatan tersebut membutuhkan waktu komputasi tinggi dan belum tentu menghasilkan konfigurasi optimal, terutama pada klasifikasi multi-kelas dengan karakter visual yang saling mirip [9].

Peningkatan performa model saat ini banyak dicapai melalui kombinasi arsitektur ringan dan optimasi hyperparameter otomatis. *Lightweight* CNN dikembangkan dengan komponen yang lebih efisien sehingga mampu menekan kompleksitas komputasi tanpa mengurangi akurasi secara signifikan [10]. Di sisi lain, *Bayesian Optimization* menjadi metode yang efektif untuk menemukan kombinasi hyperparameter terbaik secara adaptif melalui model probabilistik dengan jumlah evaluasi lebih sedikit dibanding *grid search* maupun *random search*. Pendekatan ini telah banyak digunakan untuk meningkatkan performa CNN pada berbagai kasus klasifikasi citra dan prediksi data kompleks [11].

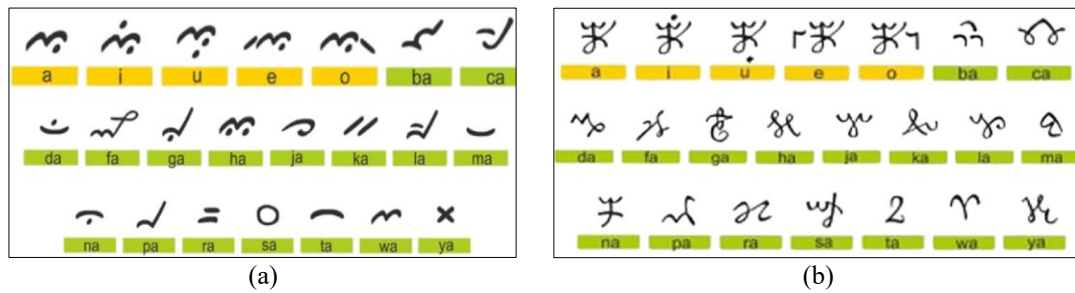
Berdasarkan kondisi tersebut, penelitian ini bertujuan mengoptimasi kinerja arsitektur *lightweight* CNN menggunakan pendekatan Bayesian untuk identifikasi aksara Bima. Kebaruan penelitian terletak pada integrasi CNN ringan yang efisien dengan optimasi hyperparameter sistematis pada dataset 44 kelas karakter yang mencakup aksara Bima lama dan baru. Ruang lingkup penelitian meliputi penyusunan *dataset*, *preprocessing* citra, augmentasi data, perancangan model *lightweight* CNN, optimasi menggunakan Optuna berbasis *Tree-structured Parzen Estimator*, serta evaluasi menggunakan *accuracy*, *precision*, *recall*, dan *F1-score*. Model yang dihasilkan diharapkan akurat, ringan, dan sesuai untuk implementasi pada perangkat komputasi terbatas sekaligus mendukung pelestarian digital aksara daerah [12], [13].

II. Metode

A. Persiapan Dataset

Dataset dalam penelitian ini dikonstruksi melalui dua metode akuisisi utama demi memperoleh representasi karakter yang menyeluruh. Tahap pertama melibatkan proses digitalisasi terhadap fragmen manuskrip kuno guna merekam karakteristik visual historis dari skrip tersebut. Sementara itu, tahap kedua memanfaatkan platform Google Formulir secara digital untuk menghimpun sampel tulisan tangan kontemporer langsung dari penutur asli (*native speaker*) masyarakat Bima. Melalui integrasi kedua pendekatan ini, diperoleh total 6.190 citra yang didistribusikan ke dalam 44 kelas karakter, yang memuat ragam tulisan aksara Bima lama (Tunti Nggahi Mbojo) maupun aksara Bima baru (Tunti Bou).

Tingginya diversitas bentuk visual dari kedua varian skrip tersebut memegang peranan krusial dalam memperkaya fase pembelajaran model serta mengoptimalkan kapabilitas generalisasinya saat memproses data baru [14]. Meskipun demikian, kuantitas sampel pada masing-masing kelas memiliki kecenderungan tidak seimbang (*imbalanced dataset*), dengan jumlah data yang berfluktuasi antara 140 hingga 196 citra per kategori karakter. Guna meminimalkan risiko bias komputasi akibat ketimpangan distribusi data tersebut, karakteristik teknis ini kemudian dimitigasi secara terstruktur melalui implementasi teknik augmentasi data. Sample dataset aksara yang digunakan pada penelitian ini ditampilkan pada Gambar 1.



Gambar 1. (a) Aksara bima baru/bou, (b) Aksara bima lama/ntoi

Kedua varian aksara memiliki perbedaan struktur visual serta terdiri atas huruf vokal dan konsonan dasar. Beberapa karakter memiliki bentuk yang mirip sehingga menuntut model mampu mengekstraksi fitur secara detail. Pendekatan deep learning seperti CNN terbukti efektif dalam mengenali pola visual kompleks melalui proses pembelajaran fitur otomatis [15].

Seluruh citra telah diberi label sesuai kelas karakter. *Dataset* kemudian dibagi menjadi data latih sebesar 70%, data validasi 15%, dan data uji 15%. Augmentasi diterapkan pada data latih untuk menambah variasi citra dan meningkatkan kualitas pembelajaran model. Skema pembagian ini digunakan untuk menjaga objektivitas evaluasi serta memastikan kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya [16].

B. Pra-Pemrosesan Gambar dan Augmentasi Data

Tahap *preprocessing* dilakukan untuk menyeragamkan kualitas, orientasi, dan format citra sebelum pelatihan model [17]. Proses awal adalah *auto-orient* untuk memastikan arah objek konsisten, karena perbedaan orientasi dapat memengaruhi ekstraksi fitur dan menurunkan performa model [18].

Citra kemudian di-resize menjadi 224×224 piksel menggunakan metode stretching agar sesuai dengan input CNN dan mengurangi kompleksitas komputasi. Transformasi resizing dinyatakan sebagai (1).

$$x' = x \cdot \frac{W_{new}}{W_{old}}, \quad y' = y \cdot \frac{H_{new}}{H_{old}} \quad (1)$$

Selanjutnya, citra dikonversi ke *grayscale* untuk menyederhanakan data menjadi satu kanal intensitas sehingga model fokus pada bentuk dan tekstur [19]. Proses ini menggunakan persamaan (2).

$$I = 0,299R + 0,587G + 0,114B \quad (2)$$

Normalisasi dilakukan dengan mengubah nilai piksel dari $[0,255]$ menjadi $[0,1]$ untuk meningkatkan stabilitas pelatihan dan mempercepat konvergensi [20] seperti persamaan (3):

$$I_{norm} = \frac{I}{255} \quad (3)$$

Untuk meningkatkan variasi data, diterapkan data *augmentation* berupa rotasi -15° hingga $+15^\circ$ guna mensimulasikan kondisi nyata dan mengurangi overfitting [21]. Proses ini menggunakan persamaan (4).

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (4)$$

Rangkaian *preprocessing* ini menghasilkan *dataset* yang lebih konsisten dan representatif sehingga meningkatkan akurasi dan efisiensi model dalam klasifikasi citra [22].

C. Arsitektur Model Lightweight CNN

Tahap klasifikasi menggunakan arsitektur *Lightweight Convolutional Neural Network* (CNN) yang dirancang untuk menghasilkan performa tinggi dengan kompleksitas komputasi rendah. Model menerima input citra hasil *preprocessing* yang telah dinormalisasi. Arsitektur yang digunakan menggabungkan *depthwise separable convolution*, *shuffle block*, *global average pooling*, serta *dense layer* dengan aktivasi *SoftMax*, sehingga mampu menjaga efisiensi tanpa mengurangi kemampuan ekstraksi fitur [23].

Proses ekstraksi fitur diawali dengan *depthwise separable convolution* yang memisahkan konvolusi menjadi dua tahap, yaitu *depthwise* dan *pointwise convolution*. Pendekatan ini mengurangi jumlah parameter secara signifikan dibandingkan konvolusi standar sehingga lebih efisien [24].

Persamaan *Depthwise* seperti (5):

$$yd(i, j, k) = \sum_{m, n} xd(i + m, j + n, k) \cdot Kd(m, n, k) \quad (5)$$

Persamaan *Pointwise* seperti (6):

$$yP(i, j, k) = \sum_c yd(i, j, c) \cdot Kp(c, k) \quad (6)$$

Selanjutnya, fitur direduksi menggunakan *Global Average Pooling* (GAP) untuk menghasilkan representasi global yang lebih ringkas dan mengurangi overfitting [25]:

$$Z_K = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W x_{ijk} \quad (7)$$

Pada tahap akhir, fitur diproses melalui dense layer dan fungsi *SoftMax* untuk menghasilkan probabilitas setiap kelas, persamaanya seperti (8):

$$P(Y_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (8)$$

dengan K sebagai jumlah kelas. Probabilitas tertinggi digunakan sebagai hasil klasifikasi. Secara keseluruhan, arsitektur ini mampu mencapai keseimbangan antara akurasi dan efisiensi komputasi, sehingga cocok untuk implementasi pada sistem real-time maupun perangkat dengan keterbatasan sumber daya.

D. Bayesian Optimization

Untuk meningkatkan performa *Lightweight CNN*, digunakan *Bayesian Optimization* karena mampu menemukan kombinasi hyperparameter optimal secara efisien pada ruang pencarian yang kompleks [26]. Proses optimasi dilakukan menggunakan framework Optuna dengan algoritma *Tree-structured Parzen Estimator* (TPE), yang memilih parameter terbaik secara adaptif berdasarkan hasil evaluasi sebelumnya [27].

Secara umum, *Bayesian Optimization* bertujuan mencari parameter optimal yang memaksimalkan fungsi objektif seperti pada persamaan (9).

$$x^* = \arg \max_{x \in X} f(x) \quad (9)$$

Pada metode TPE, distribusi parameter dibagi menjadi dua, yaitu distribusi performa baik $l(x)$ dan performa rendah $g(x)$. Parameter baru dipilih dengan memaksimalkan rasio berikut:

$$x^* = \arg \max \frac{l(x)}{g(x)} \quad (10)$$

Optimasi dijalankan selama 30 *trials*, di mana setiap trial merepresentasikan satu kombinasi hyperparameter yang diuji. Fungsi objektif menggunakan nilai akurasi validasi atau loss validasi, persamaannya seperti (11):

$$f(x) = Accuracy_{val} \text{ atau } f(x) = -Loss_{val} \quad (11)$$

Hyperparameter yang dioptimasi meliputi *learning rate*, *batch size*, jumlah *filter*, jumlah *layer*, dan *dropout* sebagaimana ditunjukkan pada Tabel 1.

Tabel 1. Ruang Pencarian Hyperparameter

Parameter	Rentang Nilai	Fungsi
Learning rate	1e-5 – 1e-2	Mengatur kecepatan pembelajaran
Batch size	16 – 128	Menentukan jumlah data per iterasi
Jumlah filter	16 – 128	Mengontrol kapasitas fitur
Jumlah layer	2 – 5	Menentukan kedalaman model
Dropout	0.2 – 0.5	Mengurangi overfitting

Pendekatan *Bayesian Optimization* berbasis TPE membuat proses pencarian hyperparameter lebih sistematis, efisien, dan mampu menghasilkan konfigurasi model terbaik dengan jumlah eksperimen yang lebih sedikit [28].

E. Evaluation Metrics

Evaluasi model dilakukan untuk mengukur kemampuan sistem dalam mengklasifikasikan citra secara akurat dan konsisten [29]. Metrik yang digunakan meliputi accuracy, precision, recall, F1-score, dan confusion matrix yang ditunjukkan pada persamaan (12) – (15).

Accuracy mengukur proporsi prediksi benar terhadap seluruh data uji.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Precision mengukur ketepatan prediksi positif, sedangkan recall mengukur kemampuan model mendeteksi data positif.

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

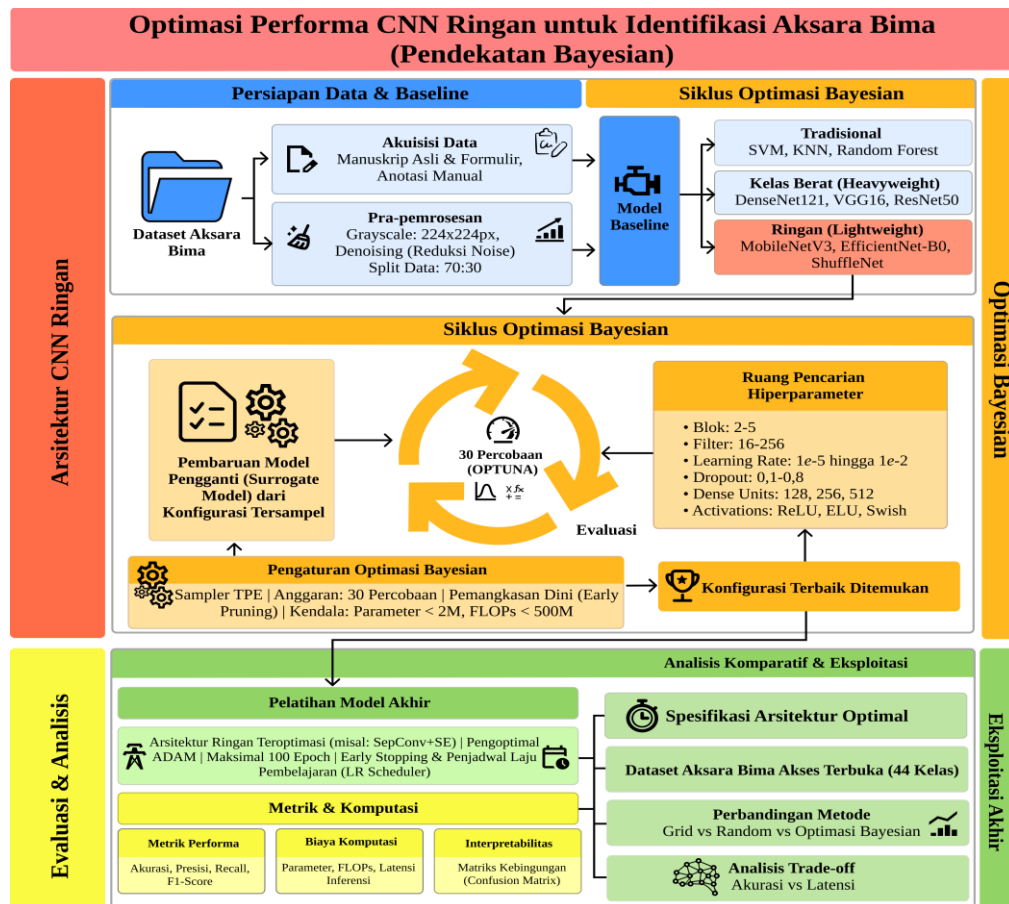
F1-score digunakan untuk menyeimbangkan precision dan recall.

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

Confusion matrix digunakan untuk menampilkan distribusi prediksi TP, TN, FP, dan FN sehingga pola kesalahan model dapat dianalisis lebih rinci [30].

F. Rancangan penelitian

Penelitian ini dirancang melalui serangkaian tahapan yang meliputi pengumpulan *dataset* aksara Bima, *preprocessing* citra, pengembangan arsitektur *lightweight* CNN, optimasi hyperparameter menggunakan *Bayesian Optimization*, dan evaluasi performa model. Kerangka kerja penelitian secara umum ditunjukkan pada Gambar 2.



Gambar 2. Rancangan Penelitian

III. Hasil dan Pembahasan

A. Distribusi Dataset dan Augmentasi Data

Dataset penelitian ini terdiri dari 6.190 citra aksara Bima yang mewakili 44 kelas karakter. Data kemudian dibagi menjadi subset pelatihan, validasi, dan pengujian untuk mendukung proses pembelajaran serta evaluasi model secara objektif.

Untuk meningkatkan jumlah data latih dan memperkaya variasi bentuk karakter, diterapkan augmentasi pada subset pelatihan melalui rotasi, translasi, dan perubahan skala citra. Teknik ini membantu model mengenali variasi pola tulisan serta meningkatkan kemampuan generalisasi.



Gambar 3. Contoh Hasil Augmentasi Data, dari kiri ke kanan: (a) citra asli, (b) rotasi, (c) translasi, dan (d) zoom

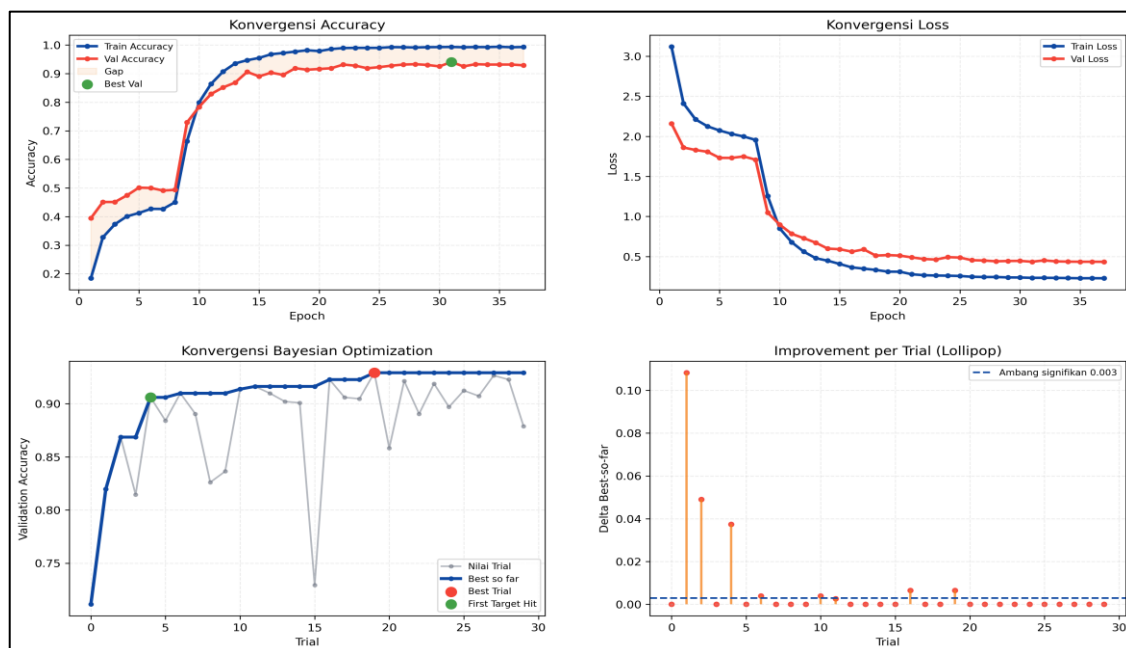
Setelah augmentasi, jumlah data latih meningkat menjadi 8.057 citra, sedangkan data validasi dan data uji tetap digunakan sebagai acuan evaluasi. Distribusi dataset ditunjukkan pada Tabel 2.

Tabel 2. Distribusi Dataset Penelitian

Subset	Jumlah
Data Asli	6.190
Data Latih (augmentasi)	8.057
Validasi	776
Uji	389

B. Analisis Konvergensi Pelatihan Model

Proses pelatihan model usulan dianalisis menggunakan kurva akurasi dan loss pada data pelatihan maupun validasi. Hasil pelatihan menunjukkan bahwa nilai akurasi meningkat secara bertahap pada setiap epoch, baik pada data latih maupun data validasi, hingga mencapai kondisi stabil. Nilai *validation accuracy* terbaik diperoleh pada epoch ke-31 sebesar 94.07%, sedangkan nilai akhir pelatihan tercatat sebesar 92.91%. Pola ini menunjukkan bahwa model mampu mempelajari fitur citra aksara Bima secara efektif selama proses training.



Gambar 4. Kurva Konvergensi Model Berbasis Bayesian Optimization dan Lightweight CNN

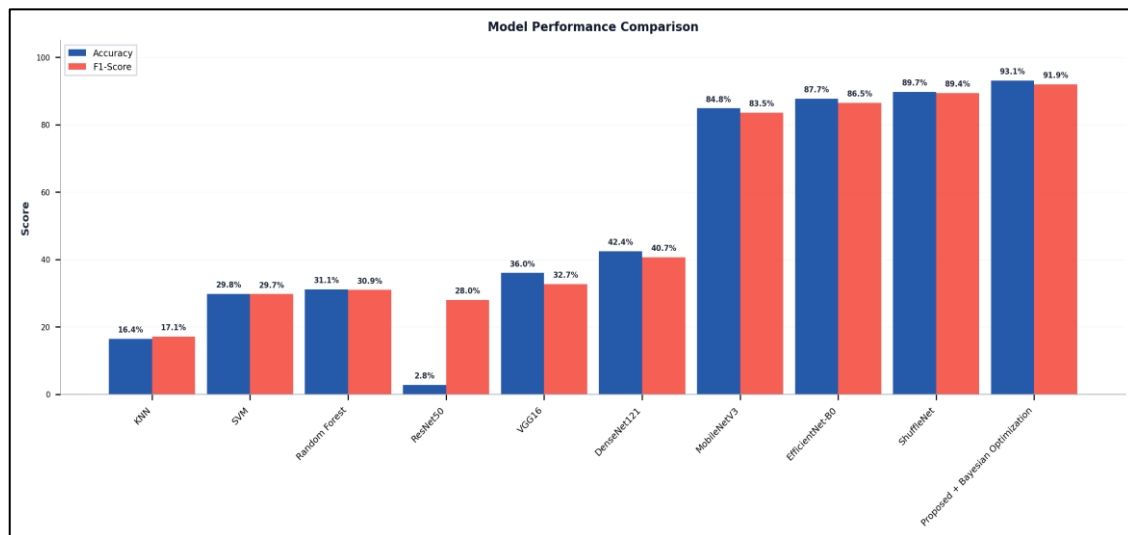
Gambar 4 menunjukkan bahwa selain peningkatan akurasi, nilai loss pada data pelatihan dan validasi mengalami penurunan secara konsisten hingga akhir epoch. Kondisi tersebut menunjukkan bahwa proses optimasi berjalan stabil dan model semakin baik dalam meminimalkan kesalahan prediksi. Selisih antara kurva pelatihan dan validasi relatif kecil, dengan generalization gap akhir sebesar 0.0640. Nilai ini menunjukkan bahwa model memiliki kemampuan generalisasi yang baik terhadap data baru dan tidak mengalami *overfitting* secara signifikan selama proses pelatihan.

C. Perbandingan Performa Model Klasifikasi

Evaluasi dilakukan untuk membandingkan kinerja metode machine learning tradisional, CNN konvensional, CNN ringan modern, serta model usulan pada data uji. Pengukuran menggunakan accuracy, precision, recall, dan F1-score untuk menilai kemampuan model dalam mengenali 44 kelas aksara Bima.

Metode tradisional masih menunjukkan performa rendah. Random Forest memperoleh accuracy tertinggi sebesar 31.11%, diikuti SVM 29.82%, dan KNN 16.45%. Hasil ini menunjukkan bahwa pendekatan berbasis fitur manual belum cukup efektif dalam menangani kompleksitas visual karakter aksara Bima.

CNN konvensional memberikan peningkatan moderat. VGG16 memperoleh accuracy 35.99% dan DenseNet121 mencapai 42.42%, sedangkan ResNet50 hanya 2.83%. Sebaliknya, CNN ringan modern menunjukkan hasil jauh lebih baik, yaitu MobileNetV3 sebesar 84.83%, EfficientNet-B0 sebesar 87.66%, dan ShuffleNet sebesar 89.72%.



Gambar 5. Perbandingan Accuracy dan F1-score Seluruh Model

Model usulan berbasis Lightweight CNN dengan Bayesian Optimization menghasilkan performa terbaik, dengan accuracy 93.06%, precision 92.26%, recall 92.55%, dan F1-score 91.91%. Hasil ini menunjukkan bahwa kombinasi arsitektur ringan dan optimasi hyperparameter mampu meningkatkan akurasi serta generalisasi model secara signifikan.

Tabel 3. Perbandingan Performa Seluruh Model

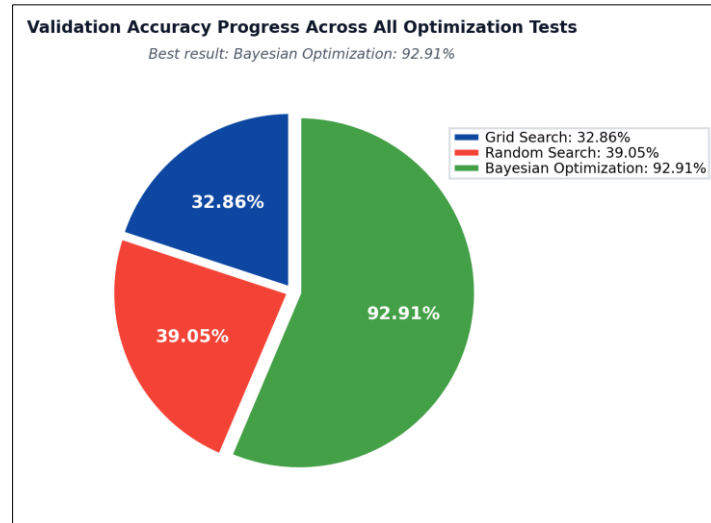
Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
KNN	16.45	24.35	16.62	17.14
SVM	29.82	33.89	29.75	29.73
Random Forest	31.11	36.92	31.3	30.95
ResNet50	2.83	0.15	2.9	0.28
VGG16	35.99	36.75	35.05	32.71
DenseNet121	42.42	43.09	42.14	40.68
MobileNetV3	84.83	83.74	85.28	83.54
EfficientNet-B0	87.66	87.13	87.05	86.5
ShuffleNet	89.72	89.79	90.78	89.4
Proposed + Bayesian Optimization	93.06	92.26	92.55	91.91

D. Analisis Optimasi Hyperparameter

Untuk menilai efektivitas strategi pencarian parameter, tiga metode optimasi dibandingkan, yaitu Grid Efektivitas pencarian hyperparameter dievaluasi dengan membandingkan tiga metode optimasi, yaitu *Grid Search*, *Random Search*, dan *Bayesian Optimization*. Perbandingan dilakukan berdasarkan jumlah percobaan serta validation accuracy terbaik yang diperoleh selama proses pencarian parameter.

Grid Search mengevaluasi 16 kombinasi parameter dan menghasilkan validation accuracy sebesar 32.86%. Meskipun dilakukan secara sistematis, metode ini kurang efisien pada ruang pencarian yang luas karena seluruh kombinasi harus diuji.

Random Search menjalankan 30 percobaan acak dan memperoleh validation accuracy sebesar 39.05%. Hasil ini lebih baik dibanding *Grid Search*, namun performanya masih bergantung pada distribusi sampel acak yang diuji.



Gambar 6. Perbandingan Validation Accuracy Metode Optimasi

Bayesian Optimization menghasilkan validation accuracy tertinggi sebesar 92.91% dalam 30 trial, dengan konfigurasi terbaik diperoleh pada trial ke-19. Target akurasi 90% telah tercapai sejak *trial* ke-4, menunjukkan proses pencarian parameter berlangsung efisien. Setelah sekitar *trial* ke-10, peningkatan performa mulai melambat, yang menandakan proses optimasi telah mendekati area parameter optimal.

Tabel 4. Perbandingan Strategi Optimasi Hyperparameter

Metode	Trials	Best Validation Accuracy (%)
Grid Search	16	32.86
Random Search	30	39.05
Bayesian Optimization	30	92.91

E. Konfigurasi Hyperparameter Optimal

Bayesian Optimization menghasilkan konfigurasi hyperparameter terbaik yang digunakan pada model usulan. Hasil optimasi memilih *backbone* MobileNetV3-Large dengan *dense units* 256, *dropout* 0.363, *head learning rate* 0.00158, *fine-tuning learning rate* 8.89×10^{-5} , *optimizer* AdamW, *label smoothing* 0.018, *unfreeze ratio* 0.5354, dan *batch size* 16.

Kombinasi parameter tersebut memberikan keseimbangan yang baik antara kecepatan konvergensi, stabilitas pelatihan, dan kemampuan generalisasi model. Penggunaan learning rate bertingkat membantu proses fine-tuning berlangsung lebih stabil, sedangkan dropout dan label *smoothing* berperan dalam mengurangi *overfitting*. Nilai unfreeze ratio menunjukkan bahwa sebagian layer backbone dibuka kembali pada tahap fine-tuning agar model dapat menyesuaikan representasi fitur secara lebih optimal terhadap dataset aksara Bima.

Tabel 5. Konfigurasi Hyperparameter Terbaik

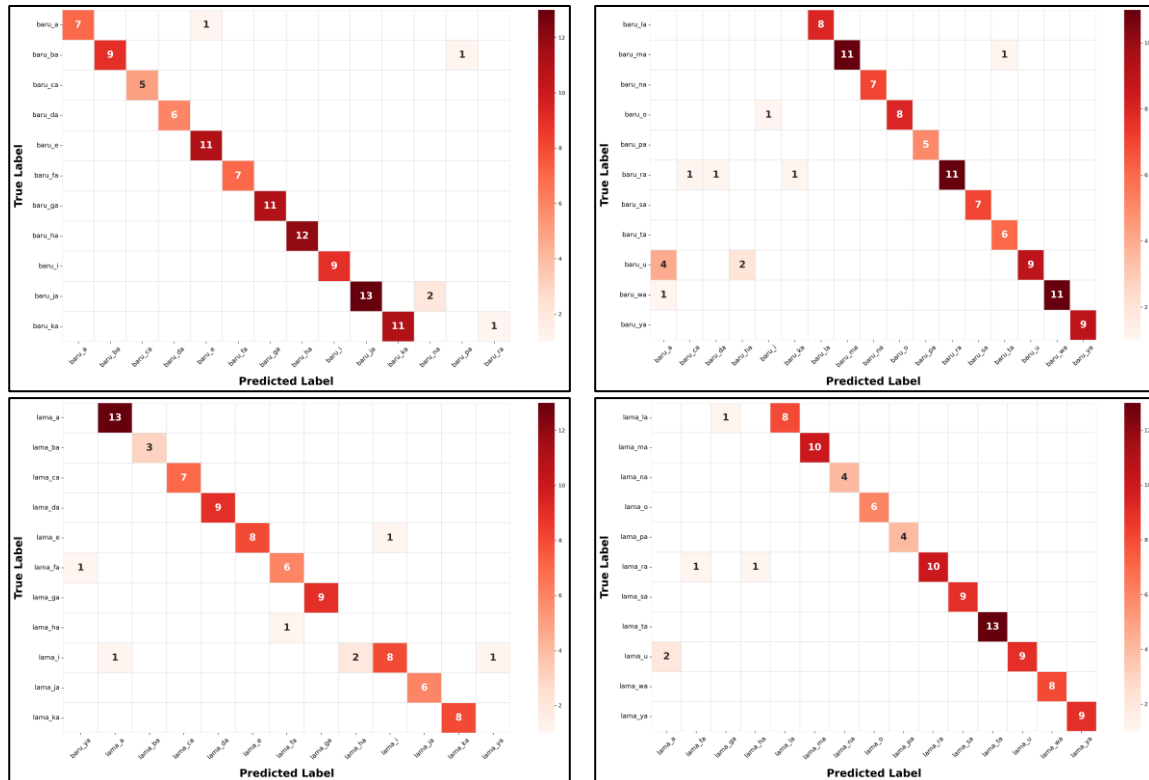
Parameter	Nilai
Backbone	MobileNetV3-Large
Dense Units	256
Dropout	0.363
Head LR	0.00158
Fine-tune LR	0.000089
Optimizer	AdamW
Label Smoothing	0.018
Unfreeze Ratio	0.5354
Batch Size	16

Konfigurasi pada Tabel 5 menunjukkan bahwa proses optimasi berhasil menemukan parameter yang sesuai untuk meningkatkan performa Lightweight CNN pada identifikasi aksara Bima.

F. Confusion Matrix dan Analisis Per Kelas

Evaluasi per kelas dilakukan menggunakan confusion matrix untuk melihat ketepatan prediksi model pada 44 kelas karakter aksara Bima. Matriks ini membantu mengidentifikasi kelas yang berhasil dikenali dengan baik maupun kelas yang masih sering tertukar.

Dominasi nilai pada diagonal utama menunjukkan bahwa sebagian besar sampel berhasil diklasifikasikan pada kelas yang benar, sehingga menandakan performa model yang tinggi dan konsisten.



Gambar 7. Confusion Matrix seluruh kelas

Beberapa kesalahan klasifikasi masih muncul pada karakter dengan bentuk visual yang mirip atau jumlah sampel yang terbatas. Namun secara umum, hasil *confusion matrix* menunjukkan bahwa model usulan mampu mengenali mayoritas kelas aksara Bima dengan baik.

G. Efisiensi Komputasi

Selain menghasilkan akurasi tinggi, model usulan juga menunjukkan efisiensi komputasi yang baik. Jumlah parameter model tercatat sebanyak 3.253.676 dengan ukuran model 36.26 MB dan ukuran bobot sebesar 12.65 MB. Kompleksitas yang relatif ringan ini mendukung proses implementasi pada perangkat dengan kapasitas komputasi terbatas.

Pengujian inferensi menunjukkan waktu pemrosesan rata-rata sebesar 63.35 ms per citra atau setara 15.79 *frame per second* (FPS). Hasil tersebut menunjukkan bahwa model mampu melakukan prediksi dengan kecepatan yang baik dan tetap layak digunakan pada sistem semi real-time maupun perangkat komputasi menengah.

Tabel 6. Evaluasi Efisiensi Komputasi Model

Metric	Nilai
Parameters	3,253,676
Model Size	36.26 MB
Weight Size	12.65 MB
Latency	63.35 ms/image
FPS	15.79

Secara keseluruhan, hasil ini menunjukkan bahwa model usulan mampu menjaga keseimbangan antara akurasi klasifikasi dan efisiensi komputasi.

H. Analisis Ablasi

Kontribusi setiap komponen arsitektur dievaluasi melalui ablation study dengan menghapus atau mengganti beberapa modul utama pada model. Pengujian ini bertujuan untuk mengukur pengaruh masing-masing komponen terhadap performa klasifikasi.

Penghapusan *Batch Normalization* menurunkan accuracy secara drastis menjadi 1.80%, menunjukkan bahwa normalisasi fitur berperan penting dalam menjaga stabilitas pelatihan dan mempercepat konvergensi. Penghapusan *SE Block* menghasilkan accuracy 22.37%, yang menandakan bahwa mekanisme perhatian kanal membantu model memfokuskan fitur penting.

Penggantian *depthwise separable convolution* dengan konvolusi standar menghasilkan accuracy 45.24%, sedangkan penghapusan *SE Block* dan *SepConv* secara bersamaan menghasilkan accuracy 55.78%. Hasil ini menunjukkan bahwa kombinasi komponen efisien pada model usulan memberikan kontribusi besar terhadap peningkatan performa.

Tabel 7. Hasil Analisis Ablasi

Configuration	Accuracy (%)
Without BatchNorm	1.80
Without SE Block	22.37
Without SepConv	45.24
Without SE + Without SepConv	55.78

Secara keseluruhan, ablation study menegaskan bahwa *Batch Normalization*, *SE Block*, dan *depthwise separable convolution* merupakan komponen penting dalam membentuk performa optimal model usulan.

I. Pembahasan

Hasil eksperimen menunjukkan bahwa metode machine learning tradisional seperti KNN, SVM, dan Random Forest masih memiliki keterbatasan pada klasifikasi aksara Bima multi-kelas. Performa yang relatif rendah menunjukkan bahwa pendekatan berbasis fitur manual belum mampu menangkap kompleksitas bentuk karakter serta kemiripan visual antar kelas. CNN konvensional seperti VGG16 dan DenseNet121 memberikan peningkatan melalui ekstraksi fitur otomatis, namun hasilnya masih berada di bawah arsitektur CNN ringan modern. Kondisi ini menegaskan bahwa rancangan jaringan yang efisien berperan penting dalam pengenalan karakter aksara dengan pola visual kompleks.

Arsitektur yang dikembangkan berbasis Lightweight CNN dan dioptimasi menggunakan Bayesian Optimization menghasilkan performa terbaik pada seluruh metrik evaluasi, dengan accuracy 93.06%, precision 92.26%, recall 92.55%, dan F1-score 91.91%. Bayesian Optimization terbukti efektif menemukan parameter terbaik secara adaptif dengan mencapai target akurasi 90% sejak trial keempat. Meskipun demikian, evaluasi kritis terhadap matriks konfusi mengungkapkan adanya *trade-off* dari penggunaan arsitektur ringan ini. Kendala pengenalan (*misclassified*) terpusat pada beberapa kelas dengan kemiripan visual tinggi. Sebagai contoh, kelas baru_a mencatat presisi rendah (0.6364) karena model sering memicu *false positive* akibat keliru mengenali aksara lama_a dan baru_ra yang memiliki kemiripan pola dasar lengkungan. Sebaliknya, kelas lama_i (recall 0.6667) dan baru_u mencatat *false negative* tinggi karena sering salah diprediksi sebagai kelas lama_fa atau baru_o, sementara kegagalan pada lama_ha murni disebabkan oleh class imbalance ekstrem (1 data uji). Pola kebingungan ini mengekspos keterbatasan arsitektur ringan, di mana operasi *depthwise separable convolution* pada MobileNetV3 mengalami reduksi parameter yang menyebabkan hilangnya sensitivitas model dalam mengekstraksi fitur spasial detail, seperti tekstur halus, titik (*diacritics*), atau lekukan minor yang menjadi ciri khas pembeda antar-aksara.

Namun secara keseluruhan, di luar keterbatasan spasial tersebut, arsitektur yang dikembangkan tetap menunjukkan efisiensi komputasi yang sangat baik melalui jumlah parameter yang relatif rendah, ukuran model yang ringkas, serta waktu inferensi yang kompetitif. Karakteristik ini menjadikan sistem mampu memberikan keseimbangan optimal antara akurasi, efisiensi komputasi, dan kemampuan generalisasi untuk identifikasi aksara Bima, sehingga sangat sesuai untuk implementasi pada perangkat bergerak maupun lingkungan dengan sumber daya terbatas.

IV. Kesimpulan dan saran

Penelitian ini berhasil mengembangkan model identifikasi aksara Bima berbasis *Lightweight Convolutional Neural Network* (CNN) yang dioptimasi menggunakan *Bayesian Optimization*. Pendekatan ini terbukti lebih unggul dibandingkan metode machine learning tradisional, CNN konvensional, maupun arsitektur *lightweight* lainnya. *Bayesian Optimization* berkinerja sangat efektif dalam mengeksplorasi ruang parameter yang

kompleks, terbukti dengan tercapainya target akurasi 90% hanya pada trial keempat. Pada pengujian terhadap 44 kelas karakter, integrasi arsitektur ringan dan optimasi adaptif ini menghasilkan model yang akurat, stabil, dan efisien secara komputasi (memiliki 3.253.676 parameter dengan waktu inferensi 63.35 ms per citra), serta mencatatkan performa klasifikasi yang tinggi dengan accuracy 93.06%, precision 92.26%, recall 92.55%, dan F1-score 91.91%. Hasil ini menegaskan bahwa model usulan sangat relevan untuk klasifikasi citra beresolusi kompleks dengan jumlah data terbatas, serta berpotensi besar untuk diterapkan pada digitalisasi manuskrip, pelestarian budaya lokal, media pembelajaran, hingga sistem OCR pada mobile dan edge devices. Meskipun demikian, evaluasi kritis menunjukkan adanya *trade-off* pada penggunaan arsitektur lightweight. Reduksi parameter pada model ini menyebabkan hilangnya sebagian detail spasial minor, yang memicu terjadinya misklasifikasi pada beberapa kelas aksara dengan tingkat kemiripan visual yang tinggi. Oleh karena itu, penelitian selanjutnya perlu diarahkan pada perluasan dataset dari berbagai sumber manuskrip dan peningkatan kualitas citra historis. Untuk mengatasi keterbatasan ekstraksi fitur halus, integrasi mekanisme atensi (attention mechanism), arsitektur berbasis transformer, atau sistem multimodal sangat direkomendasikan guna meningkatkan sensitivitas model. Selain itu, pengembangan sistem deteksi dan segmentasi karakter otomatis, serta implementasi model secara langsung pada perangkat mobile akan menjadi langkah penting dalam menyempurnakan sistem identifikasi aksara daerah di masa mendatang.

Ucapan Terima Kasih

Penulis menyampaikan terima kasih kepada semua pihak yang telah memberikan dukungan dan bantuan dalam pelaksanaan penelitian ini. Penghargaan khusus disampaikan kepada orang tua dan keluarga atas doa, motivasi, serta dukungan yang diberikan selama proses studi dan penyusunan artikel ini. Terima kasih juga kepada pembimbing, institusi, dan rekan-rekan yang telah memberikan arahan serta masukan yang bermanfaat.

Daftar Pustaka

- [1] H. Assemblali, S. Bouhsissin, and N. Sael, "Green Energy and Intelligent Transportation Deep learning-driven CNN model for detection and classification of dynamic obstacles," *Green Energy Intell. Transp.*, vol. 5, no. 3, p. 100334, 2026, doi: 10.1016/j.geits.2025.100334.
- [2] N. Paul, G. C. Sunil, D. Horvath, and X. Sun, "Deep learning for plant stress detection: A comprehensive review of technologies, challenges, and future directions," *Comput. Electron. Agric.*, vol. 229, no. December 2024, p. 109734, 2025, doi: 10.1016/j.compag.2024.109734.
- [3] A. A. Handoko, "Implementasi Convolutional Neural Network (CNN) Untuk Pengenalan Tulisan Tangan Aksara Bima," vol. 14, no. 1, pp. 96–110, 2024, doi: 10.32664/smatika.v14i01.1196.
- [4] F. F. Imron Ichwani, "Sejarah Perkembangan Ilmu Pengetahuan Dan Teknologi Dalam Islam," *UJJI*, vol. 13, no. No. 2, pp. 313–326, 2023, doi: <https://doi.org/10.47200/ulumuddin.v13i2.2085>.
- [5] P. Karthika and M. Premkumar, "Advances in Biomarker Sciences and Technology An in-depth exploration of CNN-based deep learning models in cervical carcinoma analysis," *Adv. Biomark. Sci. Technol.*, vol. 8, no. October 2025, pp. 19–33, 2026, doi: 10.1016/j.abst.2025.11.007.
- [6] R. Raju and T. M. Thasleema, "Franklin Open BO-CNN: A deep learning framework with Bayesian hyperparameter tuning for nutrient stress classification in Piper nigrum," *Franklin Open*, vol. 13, no. October, p. 100444, 2025, doi: 10.1016/j.fraope.2025.100444.
- [7] D. Xinjie *et al.*, "ScienceDirect A Lightweight Deep Learning Model for Aerial Image Classification A Lightweight Deep Learning Model for Aerial Image Classification and Its Application to UAV-based Disaster Management and Its Application to UAV-based Disaster Management," 2024, doi: 10.1016/j.procs.2024.09.395.
- [8] P. Mishra, "Chemometrics and Intelligent Laboratory Systems A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks," vol. 223, no. February, 2022, doi: 10.1016/j.chemolab.2022.104520.
- [9] T. Zheng, X. Yang, J. Lv, M. Li, S. Wang, and W. Li, "Engineering Science and Technology, an International Journal An efficient mobile model for insect image classification in the field pest management," *Eng. Sci. Technol. an Int. J.*, vol. 39, p. 101335, 2023, doi: 10.1016/j.jestch.2023.101335.
- [10] A. Syauqi and A. Prasetya, "Advances in machine transliteration methods, limitations, challenges, applications and future directions," *Nat. Lang. Process. J.*, vol. 11, no. December 2024, p. 100158, 2025, doi: 10.1016/j.nlp.2025.100158.
- [11] Y. Wang, Y. Su, Z. Zheng, Z. Zhou, and X. Wang, "Developments in the Built Environment Bayesian-optimized CNN-LSTM neural network for predicting road construction dust concentrations," *Dev. Built Environ.*, vol. 25, no. October 2025, p. 100843, 2026, doi: 10.1016/j.dibe.2026.100843.
- [12] A. Allazem and E. Mohamedelhassan, "Geodata and AI A hybrid deep learning-Bayesian optimization model for enhanced slope stability classification," *Geod. AI*, vol. 5, no. July, p. 100030, 2025, doi: 10.1016/j.geoai.2025.100030.
- [13] M. Joy, M. Santhosh, T. Park, and R. Mallipeddi, "Heliyon Bayesian optimization of convolutional

- neural network for identifying red wines using colorimetric sensors,” *Heliyon*, vol. 12, no. 2, p. e44387, 2026, doi: 10.1016/j.heliyon.2025.e44387.
- [14] J. Omor, M. Abu, T. Rony, and M. Shariful, “BNVGLENET : Hypercomplex Bangla handwriting character recognition with hierarchical class expansion using Convolutional Neural Networks,” *Nat. Lang. Process. J.*, vol. 7, no. June 2023, p. 100068, 2024, doi: 10.1016/j.nlp.2024.100068.
- [15] T. Saleh *et al.*, “Knowledge-Based Systems A vision transformer-based hybrid neural architecture for automated handwritten Bangla character recognition and braille conversion,” *Knowledge-Based Syst.*, vol. 330, no. PB, p. 114546, 2025, doi: 10.1016/j.knosys.2025.114546.
- [16] M. S. Elhadidy, A. T. Elgohr, A. Mousa, A. Safwat, R. Ismail, and H. M. Kasem, “Results in Engineering Benchmarking Pre-trained CNNs and Vision Transformers for Mpox-related Dermatological Image Classification on MSLD v2 . 0,” *Results Eng.*, vol. 28, no. October, p. 108071, 2025, doi: 10.1016/j.rineng.2025.108071.
- [17] M. Faisal *et al.*, “Utilizing Machine Learning-Based Decision-Making to Align Higher Education Curriculum with Industry Requirements,” *Int. J. Mod. Educ. Comput. Sci.*, vol. 17, no. 4, pp. 1–25, Aug. 2025, doi: 10.5815/ijmecs.2025.04.01.
- [18] L. Kang, X. Fu, L. Gomez, A. Fornés, and E. Valveny, “Preserving privacy without compromising accuracy : Machine unlearning for handwritten text recognition,” *Pattern Recognit.*, vol. 172, no. PA, p. 112411, 2026, doi: 10.1016/j.patcog.2025.112411.
- [19] W. Xu *et al.*, “Smart Agricultural Technology Real-time pest monitoring with RSCDet : Deploying a novel lightweight detection model on embedded systems,” *Smart Agric. Technol.*, vol. 12, no. May, p. 101280, 2025, doi: 10.1016/j.atech.2025.101280.
- [20] D. Guo *et al.*, “Smart Agricultural Technology SMA-YOLO : An enhanced architecture for the detection of corn diseases based on YOLOv12,” vol. 12, no. October, 2025, doi: https://doi.org/10.1016/j.atech.2025.101502.
- [21] N. Deluxni and P. Sudhakaran, “Results in Engineering Underwater debris detection using YOLOv12 with enhanced feature extraction using R-ELAN and FlashAttention network,” *Results Eng.*, vol. 28, no. September, p. 107282, 2025, doi: 10.1016/j.rineng.2025.107282.
- [22] J. Zhu, L. Liu, W. Kou, G. Chen, and X. Li, “Smart Agricultural Technology YOLO-CDD : Real-time and accurate detection of latex collection bowl and flow states with enhanced multi-scale feature fusion,” *Smart Agric. Technol.*, vol. 12, no. August, p. 101305, 2025, doi: 10.1016/j.atech.2025.101305.
- [23] V. G. Dhanya *et al.*, “Artificial Intelligence in Agriculture Deep learning based computer vision approaches for smart agricultural applications,” *Artif. Intell. Agric.*, vol. 6, pp. 211–229, 2022, doi: 10.1016/j.aiia.2022.09.007.
- [24] F. Olubusola and M. Olusoji, “Heliyon Deep learning and content-based filtering techniques for improving plant disease identification and treatment recommendations : A comprehensive review,” *Heliyon*, vol. 10, no. 9, p. e29583, 2024, doi: 10.1016/j.heliyon.2024.e29583.
- [25] Y. Saini, V. Somani, and N. Khatri, “Critical Analysis of Advanced Machine Learning and Deep Learning Models for Crop Disease Detection: Challenges, Innovations, and Future Directions,” *Franklin Open*, p. 100584, 2026, doi: 10.1016/j.fraope.2026.100584.
- [26] J. A. Paulson and C. Tsay, “ScienceDirect Bayesian optimization as a flexible and efficient design framework for sustainable process systems,” *Curr. Opin. Green Sustain. Chem.*, vol. 51, p. 100983, 2025, doi: 10.1016/j.cogsc.2024.100983.
- [27] T. Kawabata, T. Tsuzuki, T. Tatsukawa, and K. Matsui, “Allergology International Black-box optimization in immunology and beyond : A practical guide to algorithms and future directions,” *Allergol. Int.*, vol. 74, no. 4, pp. 549–562, 2025, doi: 10.1016/j.alit.2025.08.006.
- [28] X. Wen, X. Guo, S. Wang, Z. Lu, and Y. Zhang, “Biocybernetics and Biomedical Engineering Breast cancer diagnosis : A systematic review,” *Biocybern. Biomed. Eng.*, vol. 44, no. 1, pp. 119–148, 2024, doi: 10.1016/j.bbe.2024.01.002.
- [29] H. Deshpande, D. Chaudhari, T. Sarode, A. Kamath, and D. Khan, “ScienceDirect ScienceDirect LeanConv-SENet : A A novel novel lightweight lightweight Neural Neural Network Network Architecture for for Handwritten Handwritten character character recognition recognition Architecture,” *Procedia Comput. Sci.*, vol. 258, pp. 2826–2836, 2025, doi: 10.1016/j.procs.2025.04.543.
- [30] M. Jumarlis *et al.*, “A hybrid hue saturation lightness, gray level co-occurrence matrix, and k-nearest neighbour for palm-sugar classification,” *IAES Int. J. Artif. Intell.*, vol. 13, no. 3, p. 2934, Sep. 2024, doi: 10.11591/ijai.v13.i3.pp2934-2945.