

Evaluasi Kinerja Arsitektur *Lightweight GhostNet* pada Klasifikasi Tulisan Tangan Aksara Batak

Egi Dio Bagus Sudewo^{1*}, Hainur Rasyid², Muhammad Kunta Biddinika³, Kariyamin⁴,
Reni Yunita⁵,

^{1,2,5}Program Studi Sistem Informasi, Universitas Royal

³Program Studi Magister Informatika, Universitas Ahmad Dahlan

⁴Program Studi Teknologi Informasi, Institut Teknologi dan Bisnis Muhammadiyah Wakatobi

E-mail: ^{1*}egidio8gali@gmail.com, ²hainurrasyid@royal.ac.id,

³muhammad.kunta@mti.uad.ac.id, ⁴karyaminyamin28@gmail.com, ⁵reniyunita@royal.ac.id

Abstrak

Aksara Batak merupakan salah satu warisan budaya tertulis Indonesia yang keberadaannya semakin terancam akibat menurunnya penggunaan serta terbatasnya sumber daya digital untuk pembelajaran dan pelestariannya. Digitalisasi melalui teknologi pengenalan karakter otomatis menjadi salah satu solusi yang dapat mendukung upaya preservasi aksara tersebut. Namun, klasifikasi tulisan tangan Aksara Batak menghadapi tantangan berupa tingginya variasi bentuk karakter, kemiripan visual antaraksara, serta jumlah kelas yang besar. Penelitian ini bertujuan untuk mengevaluasi kinerja arsitektur *lightweight GhostNetv2_100* dalam klasifikasi tulisan tangan Aksara Batak. Dataset yang digunakan terdiri atas 76.050 citra yang terbagi ke dalam 117 kelas, dengan skema pembagian data 70% data latih, 20% data validasi, dan 10% data uji. Tahap pra-pemrosesan meliputi konversi grayscale, thresholding adaptif, resize citra menjadi 224×224 piksel, normalisasi piksel, dan one-hot encoding label. Model dilatih menggunakan optimizer Adam, batch size 64, dan maksimum 30 epoch. Hasil eksperimen menunjukkan bahwa *GhostNetv2_100* mampu mencapai akurasi sebesar 99,34%, precision sebesar 99,36%, recall sebesar 99,34%, dan F1-score sebesar 99,34% pada data uji. Dengan total parameter sebesar 5,02 juta (19,17 MB), beban komputasi hanya 0,02 GFLOPs, serta latensi inferensi rata-rata 33,74 ms per citra pada CPU, temuan ini membuktikan efisiensi *GhostNetv2_100* untuk implementasi sistem klasifikasi Aksara Batak pada perangkat mobile.

Kata kunci - Aksara Batak, klasifikasi citra, deep learning, convolutional neural network, *GhostNetv2*.

1. PENDAHULUAN

Pelestarian warisan budaya tertulis merupakan bagian penting dalam menjaga identitas, sejarah, dan pengetahuan suatu bangsa. Salah satu warisan budaya Indonesia yang memiliki nilai historis tinggi adalah Aksara Batak, yaitu sistem tulisan tradisional masyarakat Batak di Sumatra Utara yang mencakup varian Toba, Karo, Pakpak, Simalungun, Angkola, dan Mandailing [1]. Aksara ini banyak ditemukan pada media tradisional seperti bambu, kayu, tulang, dan manuskrip kuno Pustaha Laklak [2]. Namun, penggunaan Aksara Batak semakin menurun akibat dominasi alfabet Latin dan terbatasnya media pembelajaran digital, sehingga keberadaannya menghadapi ancaman kepunahan [3]. Kondisi tersebut mendorong perlunya upaya digitalisasi dan pelestarian berbasis teknologi untuk menjaga keberlangsungan aksara tersebut.

Digitalisasi tulisan tangan Aksara Batak menghadapi berbagai tantangan karena karakter yang ditulis secara manual memiliki variasi bentuk yang tinggi. Perbedaan gaya penulisan, ketebalan goresan, kemiringan karakter, serta degradasi media penyimpanan dapat menghasilkan

noise visual yang mempersulit proses pengenalan karakter [4]. Selain itu, kemiripan visual antarhuruf dan keberadaan tanda diakritik yang memodifikasi vokal menyebabkan tingkat ambiguitas yang tinggi pada proses klasifikasi [5]. Kompleksitas tersebut menuntut penggunaan metode pengenalan pola yang mampu mengekstraksi fitur secara otomatis dan robust terhadap variasi tulisan tangan.

Perkembangan teknologi computer vision dan deep learning telah memberikan solusi yang efektif melalui penerapan *Convolutional Neural Network* (CNN). CNN mampu mempelajari representasi fitur secara otomatis mulai dari pola sederhana hingga struktur karakter yang kompleks tanpa memerlukan ekstraksi fitur manual [6][7]. Meskipun demikian, banyak arsitektur CNN modern memiliki jumlah parameter dan kebutuhan komputasi yang besar sehingga kurang sesuai untuk implementasi pada perangkat bergerak dan sistem tertanam yang memiliki keterbatasan sumber daya [8], [9], [10]. Oleh karena itu, diperlukan arsitektur yang tidak hanya akurat tetapi juga efisien dari sisi memori dan komputasi.

Beberapa penelitian terdahulu telah menunjukkan potensi *deep learning* dalam pengenalan aksara tradisional, namun masih menyisakan keterbatasan metode. Penelitian oleh Wijaya dan Via [11] mengombinasikan CNN dan *Extreme Learning Machine* untuk klasifikasi Aksara Bali dan memperoleh akurasi sebesar 91%, namun masih menggunakan dataset yang relatif kecil sehingga ketahanan model terhadap variasi goresan yang lebih kompleks belum teruji. Mulyanto dkk. [12] menerapkan CNN untuk pengenalan tulisan tangan Aksara Lampung berbasis OCR dan memperoleh presisi 87%, meskipun akurasi pelatihannya masih rendah (57%) yang mengindikasikan generalisasi model belum optimal. Rahim dkk. [13] membandingkan beberapa arsitektur CNN pada klasifikasi Aksara Lontara dan menemukan bahwa VGG16 mencapai akurasi terbaik sebesar 90%, tetapi model tersebut memiliki ukuran bobot ratusan megabita dan membutuhkan sumber daya komputasi tinggi. Sementara itu, Guo dkk. [14] mengembangkan model *multi-branch deep CNN* untuk pengenalan aksara kuno dengan akurasi 99,94%, namun memiliki kompleksitas struktural yang besar.

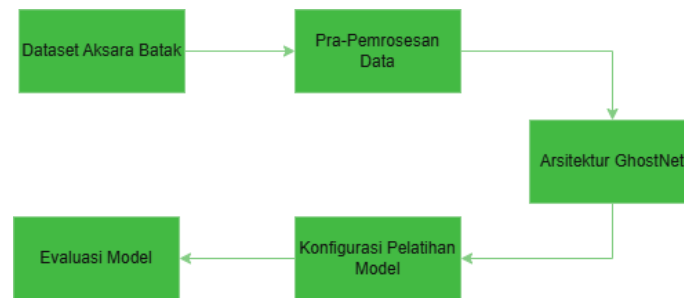
Untuk mengatasi tantangan komputasi pada perangkat terbatas, beberapa arsitektur *lightweight* seperti MobileNet dan EfficientNet umum diajukan. Namun, MobileNet dengan *depthwise separable convolution* murni terkadang kehilangan representasi fitur detail halus pada karakter kecil, sedangkan EfficientNet memiliki kedalaman komputasi yang kurang fleksibel dalam menangkap korelasi spasial tanda diakritik. Di sisi lain, Han dkk. [15] memperkenalkan GhostNet yang mampu memangkas FLOPs dan parameter secara drastis melalui mekanisme *cheap linear operations*. Pengembangan lanjutannya, GhostNetv2_100, mengintegrasikan mekanisme *Decoupled Fully Connected* (DFC) Attention yang sangat efektif menangkap hubungan spasial jarak jauh secara linear. Mekanisme ini sangat relevan untuk mengenali keterkaitan antara karakter induk (*ina ni surat*) dan tanda diakritik (*anak ni surat*) pada Aksara Batak tanpa membebani memori kerja perangkat.

Berdasarkan kajian tersebut, masih terdapat kesenjangan penelitian terkait penerapan arsitektur *lightweight* ber-atensi spasial untuk klasifikasi tulisan tangan Aksara Batak dengan jumlah kelas berskala besar. Sebagian besar penelitian terdahulu masih berfokus pada aksara daerah lain atau menggunakan model dengan kompleksitas tinggi [16], [17], [18]. Oleh karena itu, penelitian ini berfokus mengevaluasi kinerja arsitektur GhostNetv2_100 dalam mengklasifikasikan 117 kelas tulisan tangan Aksara Batak yang terdiri atas 76.050 citra. Penelitian ini bertujuan menganalisis akurasi serta efisiensi komputasi model, sehingga dapat memberikan landasan empiris bagi pengembangan aplikasi pelestarian budaya berbasis *on-device mobile AI*.

2. METODE PENELITIAN

Metode penelitian yang digunakan dalam studi ini terdiri atas beberapa tahapan utama, yaitu pengumpulan dan pembagian dataset, pra-pemrosesan citra, perancangan model GhostNetv2_100, proses pelatihan model, serta evaluasi kinerja menggunakan metrik akurasi, precision, recall, dan F1-score. Alur penelitian pada Gambar 1 dirancang untuk mengukur

kemampuan arsitektur lightweight GhostNetv2_100 dalam mengklasifikasikan tulisan tangan Aksara Batak secara akurat dan efisien pada dataset berskala besar.



Gambar 1. Alur Penelitian

2.1 Dataset Aksara Batak

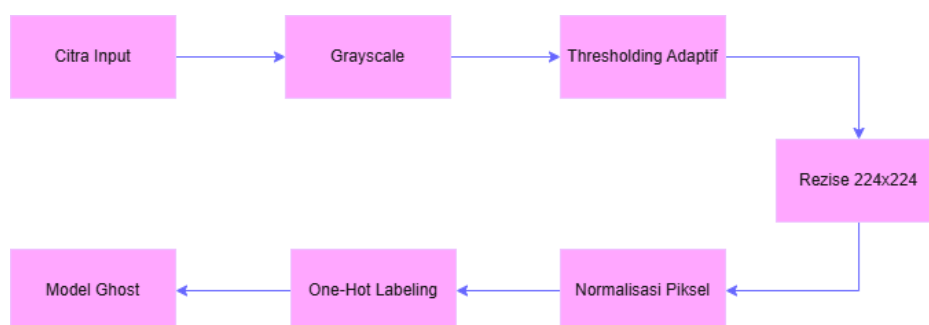
Dataset tulisan tangan Aksara Batak yang digunakan dalam penelitian ini merupakan representasi komprehensif dari aksara tradisional Batak. Data dikumpulkan dari sampel tulisan tangan asli siswa di daerah Sumatra Utara untuk menangkap variasi goresan riil. Struktur Aksara Batak terdiri atas 19 karakter induk (*ina ni surat*) dan kombinasi diakritik (*anak ni surat*) yang secara keseluruhan membentuk 117 kelas karakter unik. Pengumpulan sampel dilakukan dengan variasi alat tulis (pena standar dan spidol), media kertas bergaris maupun polos, serta rentang kecepatan penulisan alami siswa guna menghasilkan diversitas gaya penulisan. Total citra adalah sebanyak 76.050 citra biner berformat PNG yang dibagi secara acak dengan rasio 70% data latih (53.313 citra), 20% data validasi (15.231 citra), dan 10% data uji (7.622 citra) sebagaimana disajikan pada Tabel 1.

Tabel 1. Pembagian Dataset

Kategori Data	Persentase	Jumlah Citra
Latih (<i>Training</i>)	70%	53,313
Validasi (<i>Validation</i>)	20%	15,231
Uji (<i>Testing</i>)	10%	7,622
Total	100%	76,050

2.2 Pra-pemrosesan Data

Tahap pra-pemrosesan dilakukan untuk menghasilkan data masukan yang seragam dan siap digunakan oleh model GhostNet. Proses ini diawali dengan konversi citra ke format grayscale dan penerapan thresholding adaptif untuk memisahkan objek tulisan dari latar belakang. Selanjutnya, seluruh citra diubah ukurannya menjadi 224×224 piksel sesuai dengan dimensi masukan standar GhostNet. Nilai piksel kemudian dinormalisasi ke rentang 0–1 dan distandarisasi menggunakan nilai mean serta standar deviasi ImageNet guna meningkatkan stabilitas proses pelatihan. Pada tahap akhir, label dari 117 kelas dikonversi menjadi representasi one-hot encoding untuk mendukung perhitungan fungsi *loss Categorical Cross-Entropy* [19]. Secara keseluruhan, alur pra-pemrosesan seperti pada Gambar 2.



Gambar 2. Alur Pra-pemrosesan Data

2.3 Arsitektur GhostNet

GhostNet merupakan arsitektur lightweight CNN yang dirancang untuk mengurangi kebutuhan komputasi dengan memanfaatkan Ghost Module, yaitu mekanisme yang menghasilkan sebagian besar peta fitur melalui operasi linier sederhana setelah sejumlah kecil peta fitur intrinsik diperoleh dari konvolusi utama [20][21]. Pendekatan ini mampu menekan jumlah parameter dan FLOPs secara signifikan dibandingkan konvolusi konvensional tanpa mengorbankan kemampuan representasi fitur [22]. Struktur utama GhostNet dibangun menggunakan Ghost Bottleneck, yang mengadopsi konsep inverted residual dengan dua Ghost Module serta *depthwise convolution* untuk efisiensi komputasi dan reduksi dimensi spasial. Pada penelitian ini digunakan varian GhostNetv2_100, yang dilengkapi mekanisme *Decoupled Fully Connected* (DFC) Attention untuk menangkap hubungan spasial jarak jauh secara efisien, sehingga berpotensi meningkatkan kemampuan model dalam mengenali keterkaitan antara huruf induk (*ina ni surat*) dan tanda diakritik (*anak ni surat*) pada Aksara Batak [23]. Secara keseluruhan, arsitektur model terdiri atas lapisan masukan berukuran $224 \times 224 \times 3$, diikuti convolution stem, tumpukan Ghost Bottleneck v2 dengan DFC Attention, Global Average Pooling, lapisan *fully connected* sebanyak 117 neuron sesuai jumlah kelas, dan lapisan Softmax sebagai keluaran klasifikasi.

2.4 Konfigurasi Pelatihan Model

Pelatihan model GhostNetv2_100 dilakukan menggunakan optimizer Adam karena kemampuannya dalam menyesuaikan learning rate secara adaptif berdasarkan estimasi momen gradien sehingga mempercepat proses konvergensi. Model dilatih dengan batch size sebesar 64 untuk menjaga keseimbangan antara stabilitas pembelajaran dan penggunaan memori GPU. Dengan proses pelatihan dievaluasi secara berkala hingga model mencapai konvergensi optimal. Untuk menangani permasalahan klasifikasi multi-kelas sebanyak 117 kategori, digunakan fungsi *loss Categorical Cross-Entropy* yang menghitung perbedaan antara distribusi probabilitas hasil prediksi dan label aktual. Seluruh eksperimen diimplementasikan menggunakan framework PyTorch melalui pustaka *timm* (Torch Image Models) untuk memanfaatkan model GhostNetv2_100 pra-terlatih, sedangkan proses pengelolaan dan pra-pemrosesan data dilakukan menggunakan TensorFlow/Keras pada lingkungan komputasi berbasis GPU.

2.5 Evaluasi Model

Evaluasi kinerja model dilakukan menggunakan data uji yang tidak terlibat dalam proses pelatihan maupun validasi untuk mengukur kemampuan generalisasi GhostNetv2_100. Performa model dievaluasi menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* yang dihitung berdasarkan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Persamaan yang digunakan adalah sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

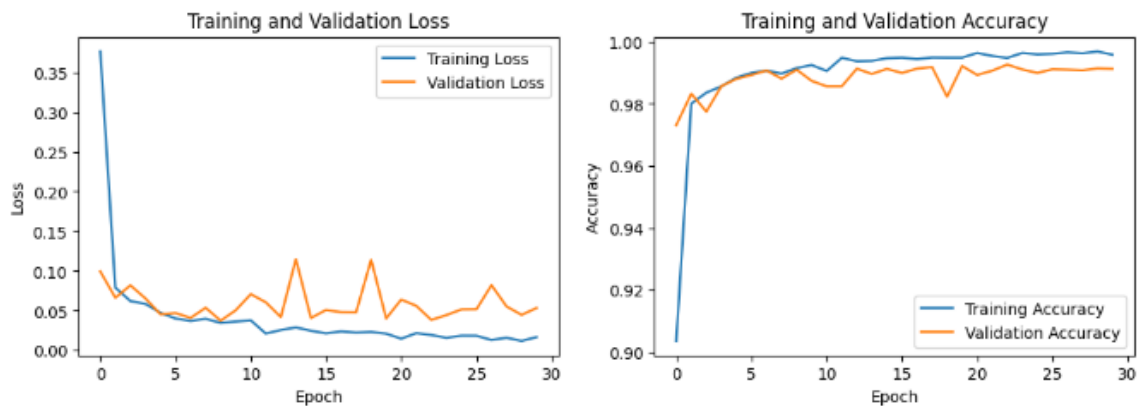
$$F1 = 2 \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Selain metrik performa klasifikasi di atas, evaluasi efisiensi komputasi model diukur melalui jumlah parameter terhitung, estimasi kebutuhan *Floating Point Operations* (FLOPs), ukuran bobot model dalam memori (*storage size*), serta waktu latensi inferensi rata-rata per citra (*inference time*) guna membuktikan kelayakan *deployment* pada perangkat *mobile*.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pelatihan Model

Proses pelatihan model GhostNetv2_100 pada dataset Aksara Batak menunjukkan grafik konvergensi yang sangat cepat dan stabil. Inisialisasi bobot pra-terlatih dari ImageNet membantu mempercepat adaptasi filter konvolusi primer pada epoch awal. Grafik akurasi pelatihan meningkat secara signifikan sejak 5 epoch pertama, yang diiringi oleh penurunan grafik loss secara tajam seperti pada Gambar 3.



Gambar 3. Grafik Training dan Validation

Pada rentang epoch ke-15 hingga ke-30, grafik akurasi validasi menunjukkan tren stabil di angka yang sangat tinggi. Tidak ditemukan adanya indikasi overfitting yang parah; hal ini ditunjukkan oleh kurva loss pelatihan dan loss validasi yang terus menurun secara beriringan hingga stabil di bawah nilai 0.02 pada epoch ke-30. Hal ini membuktikan bahwa arsitektur GhostNetv2_100 dengan kapasitas parameter yang minimal memiliki kapabilitas generalisasi yang kokoh untuk mengenali pola geometris rumit pada Aksara Batak.

3.2 Evaluasi Kinerja Model

Berdasarkan Tabel 2, model GhostNetv2_100 tidak hanya unggul dalam aspek ketepatan klasifikasi dengan perolehan akurasi 99,34% dan F1-Score 99,34%, tetapi juga menunjukkan profil efisiensi komputasi yang sangat bersahabat untuk perangkat dengan sumber daya terbatas. Kebutuhan beban komputasi model hanya sebesar 20 MFLOPs (0,02 GFLOPs) dengan konsumsi penyimpanan memori parameter sebesar 19,17 MB. Selain itu, pengujian latensi menghasilkan waktu inferensi rata-rata sebesar 33,74 ms per citra pada pemrosesan CPU tanpa bantuan akselerasi GPU, membuktikan kesiapan model untuk eksekusi secara *real-time* dan *on-device*.

Tabel 2. Hasil Evaluasi

Metrik Evaluasi	Nilai Performa	Interpretasi Kinerja
Accuracy	0.9934 (99.34%)	Tingkat ketepatan klasifikasi model sangat tinggi
Precision	0.9936 (99.36%)	Tingkat kesalahan positif palsu model sangat rendah
Recall	0.9934 (99.34%)	Kemampuan model menemukan seluruh kelas uji sangat konsisten
F1-Score	0.9934 (99.34%)	Keseimbangan metrik presisi dan recall berada pada level puncak
Model Storage Size	19.17 MB (Params Size)	Ukuran bobot file model sangat ringkas untuk memori perangkat <i>mobile</i>
Computational Load (FLOPs)	20 MFLOPs (0.02 GFLOPs)	Beban komputasi operasi floating-point sangat rendah dan ringan
Average Inference Time	33.74 ms / citra	Kecepatan inferensi pada CPU cepat dan mendukung implementasi <i>real-time</i>

Berdasarkan laporan klasifikasi (*classification report*) per kelas, mayoritas kelas karakter berhasil mencapai performa klasifikasi sempurna (nilai F1-Score 1.00). Tabel 3 menampilkan sampel sepuluh kelas berkinerja terbaik dengan F1-Score sempurna.

Tabel 3. 10 Kelas Terbaik

Nama Kelas	Presisi	Recall	F1-Score	Support
'a'	1	1	1	65
'b'	1	1	1	65
'ba'	1	1	1	65
'bang'	1	1	1	65
'be'	1	1	1	65
'bo'	1	1	1	65
'bu'	1	1	1	65
'dang'	1	1	1	65
'da'	1	1	1	65
'de'	1	1	1	65

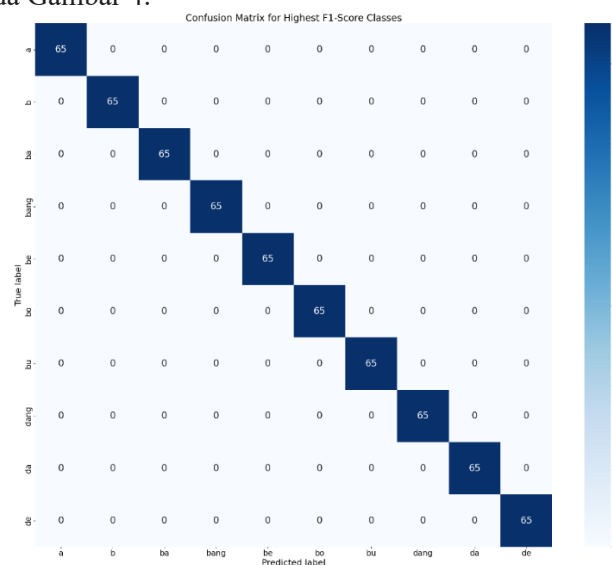
Meskipun performa global sangat tinggi, terdapat sedikit penurunan nilai F1-Score pada beberapa kelas karakter tertentu. Sepuluh kelas dengan kinerja F1-Score terendah disajikan pada Tabel 4.

Tabel 4. 10 Kelas Terendah

Nama Kelas	Presisi	Recall	F1-Score	Support
'le'	0.9697	0.9846	0.9771	65
'ne'	0.9844	0.9692	0.9767	65
'p'	0.9844	0.9692	0.9767	65
'pang'	1	0.9545	0.9767	66
'lo'	1	0.9538	0.9764	65
'hang'	0.9841	0.9538	0.9688	65
'tang'	0.9403	0.9692	0.9545	65
'te'	0.9683	0.9385	0.9531	65
'li'	0.9836	0.9231	0.9524	65
'ngi'	0.9143	0.9846	0.9481	65

3.3 Analisis Confusion Matrix

Melalui analisis detail terhadap *Confusion Matrix*, pola prediksi benar dan persebaran kesalahan klasifikasi model dapat dipetakan secara jelas. Matriks kebingungan menunjukkan diagonal yang sangat solid, mengindikasikan bahwa hampir semua citra uji terprediksi ke kelas yang tepat seperti pada Gambar 4.



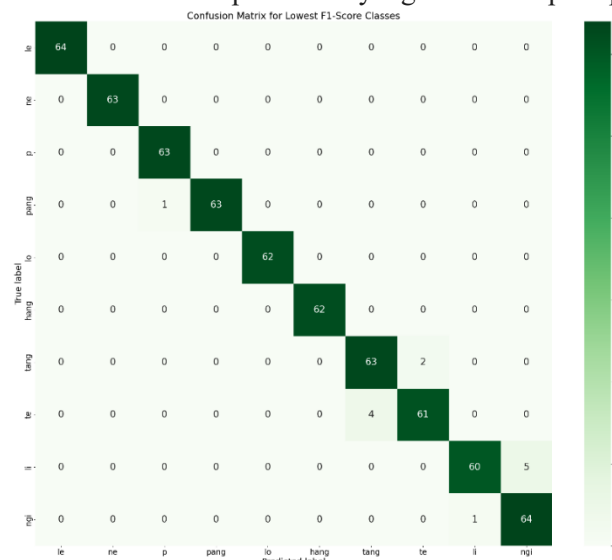
Gambar 4. Confusion Matrix 10 F1-Score Tertinggi

Pada kelas-kelas dengan recall tertinggi (nilai recall sempurna 1.00), model berhasil mengekstraksi batas geometri yang tegas dari karakter dasar. Sepuluh kelas berkinerja recall sempurna ditampilkan pada Tabel 5.

Tabel 5. 10 F1-Score Tertinggi

No. Indeks	Nama Kelas	Nilai Recall	Karakteristik Kelas
0	'a'	1	Karakter dasar tanpa diakritik kompleks
1	'b'	1	Karakter dasar dengan stroke tegas
2	'ba'	1	Pola penulisan stabil
3	'bang'	1	Struktur goresan unik
4	'be'	1	Kombinasi diakritik konsisten
6	'bo'	1	Kombinasi diakritik konsisten
7	'bu'	1	Kombinasi diakritik konsisten
8	'd'	1	Karakter dasar
9	'da'	1	Karakter dasar
21	'go'	1	Struktur goresan unik

Sebaliknya, analisis confusion matrix pada kelas dengan recall terendah mengungkap adanya hambatan klasifikasi minor akibat kemiripan visual yang ekstrem seperti pada Gambar 5.



Gambar 5. Confusion Matrix 10 F1-Score Terendah

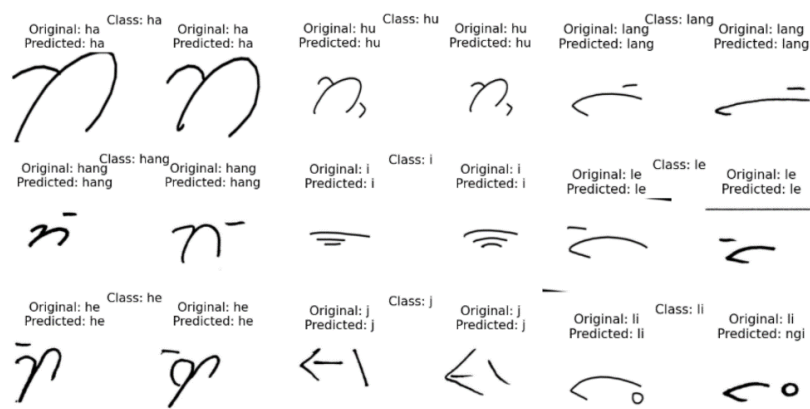
Sepuluh kelas dengan recall terendah dirangkum pada Tabel 6.

Tabel 6. 10 F1-Score Terendah

No. Indeks	Nama Kelas	Nilai Recall	Faktor Penyebab Ambiguitas
42	'li'	0.9231	Kemiripan bentuk diakritik vokal 'i'
98	'te'	0.9385	Kemiripan struktur goresan dengan diakritik 'e'
25	'hang'	0.9538	Goresan tambahan yang sangat tipis
43	'lo'	0.9538	Kemiripan posisi diakritik vokal 'o'
76	'pang'	0.9545	Kemiripan bentuk tanda panyungga
36	'jo'	0.9692	Variasi bentuk tulisan tangan individu
55	'ne'	0.9692	Goresan ekor karakter yang sangat mirip
74	'p'	0.9692	Karakter dasar berukuran kecil
83	'rang'	0.9692	Kemiripan goresan vertikal
44	'lu'	0.9692	Kemiripan bentuk diakritik vokal 'u'

Penyebab utama penurunan recall pada kelas seperti 'li' (0.9231), 'te' (0.9385), dan 'lo' (0.9538) adalah kemiripan struktural yang sangat tinggi pada varian penulisan manual. Pada tulisan tangan cepat, goresan untuk diakritik vokal 'i' ('li') sering kali bergeser posisi atau memiliki sudut kemiringan yang hampir identik dengan diakritik vokal 'o' ('lo'), sehingga membingungkan lapisan ekstraksi fitur spasial tingkat atas. Namun demikian, nilai recall terendah yang diperoleh

masih berada pada angka yang sangat tinggi (di atas 92%), menegaskan ketangguhan model. Hasil klasifikasi aksara Batak dapat dilihat pada Gambar 6.



Gambar 6. Sample Hasil Klasifikasi

3.4 Pembahasan

Keberhasilan arsitektur GhostNetv2_100 dalam mencapai akurasi uji hingga 99,34% membuktikan keandalan luar biasa dari dekomposisi *Ghost Module*. Pada tugas klasifikasi multi-kelas berskala besar (117 kelas), CNN konvensional biasanya menghasilkan ratusan peta fitur redundan melalui filter konvolusi berbiaya tinggi. GhostNet meminimalkan hal ini dengan membatasi jumlah konvolusi primer dan menggunakan operasi linier murah (*cheap linear operations*) untuk menghasilkan peta fitur bayangan. Pendekatan ini secara tidak langsung berfungsi sebagai regularisasi implisit yang mencegah model dari *overfitting*, terbukti dari kurva *loss* validasi dan latih yang menurun stabil beriringan hingga di bawah 0,02.

Efektivitas arsitektur ini diperkuat oleh integrasi perhatian *Decoupled Fully Connected* (DFC) Attention yang menangkap hubungan spasial jarak jauh antara huruf induk (*ina ni surat*) dan tanda diakritik (*anak ni surat*) secara efisien. Berdasarkan pengukuran efisiensi komputasi, GhostNetv2_100 hanya memiliki 5.025.785 parameter (5,02 M) dengan kebutuhan komputasi sebesar 0,02 GFLOPs, ukuran memori bobot parameter sebesar 19,17 MB, serta waktu inferensi rata-rata sebesar 33,74 ms per citra pada pemrosesan CPU standar. Profil komputasi yang sangat ringan ini menjamin kelayakan model untuk diintegrasikan secara *on-device* pada aplikasi *smartphone* tanpa memerlukan ketergantungan koneksi ke peladen *cloud*.

Tabel 7. Perbandingan Penelitian Terdahulu

Penulis (Tahun)	Metode dan Dataset	Hasil dan Keterbatasan
Wijaya & Via (2025)	CNN-ELM Hybrid pada 1.691 citra Aksara Bali	Mencapai akurasi uji 91%, namun ukuran dataset relatif kecil sehingga rentan terhadap variasi goresan tulisan tangan yang lebih kompleks.
Mulyanto dkk. (2021)	CNN dan OCR pada citra tulisan tangan Aksara Lampung	Memperoleh akurasi pelatihan 57% dan presisi 87%, tetapi kemampuan generalisasi model masih terbatas akibat akurasi yang rendah.
Rahim dkk. (2024)	CNN (VGG16, VGG19, ResNet, dan ResNetV2) pada 12.282 citra Aksara Lontara	VGG16 menghasilkan akurasi terbaik sebesar 90%, namun membutuhkan komputasi tinggi dan masih mengalami kesalahan pada karakter yang memiliki kemiripan visual.
Wu dkk. (2023)	MbSPPVGG (Multi-branch DCNN) pada citra buku kuno etnis minoritas Tiongkok	Mencapai akurasi dan F1-score sebesar 99,94%, tetapi arsitektur yang digunakan sangat kompleks dan memerlukan sumber daya komputasi yang besar.
Han dkk. (2020)	GhostNet pada dataset ImageNet ILSVRC-2012	Mencapai top-1 accuracy sebesar 73,9% dengan efisiensi komputasi tinggi, namun belum dievaluasi pada klasifikasi tulisan tangan dengan jumlah kelas yang besar.
Penelitian Ini	GhostNetv2_100 (76.050 citra, 117 kelas Aksara Batak)	Akurasi uji 99,34%, F1-Score 99,34%, Sangat ringan (5,02M params, 20 MFLOPs, 19,17 MB, latensi CPU 33,74 ms/citra)

Jika dibandingkan dengan penelitian terdahulu pada Tabel 7, model GhostNetv2_100 menawarkan keunggulan yang sangat dominan. Penelitian Lampung CNN oleh Mulyanto dkk. hanya mencatatkan akurasi latih sebesar 57%, sementara penelitian Lontara CNN oleh Rahim

dkk. dengan model VGG16 mencapai akurasi uji 90%. Model hybrid CNN-ELM pada aksara Bali mencapai akurasi 91%. Model CNN pada penelitian terdahulu untuk aksara Batak memang menunjukkan potensi klasifikasi yang baik, namun sebagian besar masih diuji pada ruang lingkup varian karakter yang terbatas atau dataset sintesis berbasis font komputer, sehingga belum teruji menghadapi variasi tulisan tangan asli berskala besar. Sebaliknya, penelitian ini membuktikan bahwa GhostNetv2_100 mampu menangani 117 kelas tulisan tangan riil dengan akurasi uji melampaui 99,3% menggunakan parameter model yang jauh lebih kecil dan efisien dibanding VGG16 atau ResNet.

Implikasi praktis dari penelitian ini terhadap pelestarian warisan budaya sangat signifikan. Kebutuhan daya komputasi dan memori RAM yang rendah dari GhostNetv2_100 memfasilitasi integrasi model secara langsung di dalam perangkat smartphone (*on-device AI*) tanpa membutuhkan konektivitas internet atau server cloud eksternal. Model ini dapat diterapkan sebagai modul utama pada aplikasi edukasi digital interaktif Aksara Batak, keyboard transliterasi real-time, ataupun sistem OCR pemindai manuskrip kuno Pustaka Laklak untuk mendukung pelestarian budaya secara berkelanjutan.

4. KESIMPULAN

Penelitian ini berhasil menunjukkan bahwa arsitektur *lightweight* GhostNetv2_100 sangat efektif untuk klasifikasi tulisan tangan Aksara Batak berskala besar dengan 117 kelas dan 76.050 citra, yang dibuktikan dengan perolehan akurasi uji sebesar 99,34%, presisi 99,36%, *recall* 99,34%, dan F1-score 99,34%. Model ini terbukti sangat efisien dengan parameter hanya 5,02 juta (19,17 MB), beban 20 MFLOPs, serta latensi inferensi rata-rata 33,74 ms per citra pada CPU. Keterbatasan penelitian ini terletak pada data latih yang masih bersumber dari sampel kelompok siswa di satu wilayah. Model berpotensi mengalami penurunan performa jika dihadapkan pada gaya tulisan tangan pengguna umum yang sangat acak atau manuskrip kuno (*Pustaka Laklak*) yang mengalami degradasi fisik parah.

5. SARAN

Berdasarkan hasil dan keterbatasan penelitian, pengembangan lanjutan perlu difokuskan pada tiga prioritas utama. Prioritas utama adalah meningkatkan ketahanan model terhadap kelas yang memiliki kemiripan visual tinggi melalui penerapan teknik augmentasi *elastic deformation*, rotasi acak ringan, serta perluasan variasi dataset dari kelompok penulis lintas usia dan media penulisan. Prioritas berikutnya adalah merealisasikan implementasi *on-device* pada aplikasi *mobile* interaktif berbasis Android/iOS menggunakan format teroptimasi (seperti TFLite atau ONNX) guna mengevaluasi efisiensi daya dan latensi secara *real-time*. Terakhir, penelitian selanjutnya disarankan mengembangkan sistem *Optical Character Recognition* (OCR) terpadu secara *end-to-end* yang dilengkapi modul segmentasi garis dan kata, sehingga mampu mengenali dan mentransliterasikan rangkaian kata maupun kalimat penuh Aksara Batak secara otomatis dari naskah fisik.

DAFTAR PUSTAKA

- [1] Y. Pratama, S. Try, N. Nainggolan, D. I. Nadya, and N. Y. Naipospos, "One-shot learning Batak Toba character recognition using siamese neural network," *Telkomnika*, vol. 21, no. 3, pp. 600–612, 2023, doi: 10.12928/TELKOMNIKA.v21i3.24927.
- [2] D. A. Dharmawan, B. M. Akbar, R. A. C. Leuveano, M. P. Tahya, and A. R. Yehudha, "Deep Learning Approaches for Batak Script Recognition : A Literature Review," *RSF Conf. Ser. Eng. Technol.*, vol. 4, no. 1, 2025, doi: <https://doi.org/10.31098/cset.v4i1.1069>.
- [3] E. M. Zamzami and E. Nababan, "An AI-Based Approach to Evaluate Students' Old Script Writings," *Int. J. Comput. Sci. Cult. Herit.*, vol. 12, no. 2, pp. 117–127, 2024.
- [4] M. M. Sihotang and T. Sitanggang, "Cultural Heritage Preservation and Manuscript

-
- Conservation: Safeguarding the Laklak Batak Manuscript of Batak Culture in Indonesia,” *J. Ilmu Pendidik. dan Hum.*, vol. 11, no. 2, pp. 85–101, 2022, doi: 10.35335/jiph.v11i2.17.
- [5] R. D. Sianipar, “Implementation of YOLOv8 in Real-Time Batak Script Detection,” *J. Comput. Vis. Pattern Recognit.*, vol. 8, no. 2, pp. 101–110, 2023.
- [6] O. Ediansa and I. Riadi, “Klasifikasi Jenis Penyakit Pada Buah Nanas Menggunakan Pendekatan Model VGG16 dan EfficientNetB7,” vol. 12, no. 01, pp. 42–50, 2026.
- [7] M. Yaqub *et al.*, “State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images,” *Brain Sci.*, vol. 10, no. 7, pp. 1–19, 2020, doi: 10.3390/brainsci10070427.
- [8] K. F. Habie, “Impact of Optimizer Selection on MobileNetV1 Performance for Skin Disease Detection Using Digital Images,” *JUTIF*, vol. 6, no. 3, pp. 1589–1604, 2025, doi: <https://doi.org/10.52436/1.jutif.2025.6.3.4685>.
- [9] X. Zhou, B. Huang, H. Fan, and P. Zhao, “A Lightweight Convolutional Neural Network for Face Recognition,” in *2024 International Conference on Decision Aid Sciences and Applications (DASA)*, 2024, pp. 1–5. doi: 10.1109/DASA63652.2024.10836448.
- [10] O. S. Parulian, “Efficient Design and Compression of CNN Models for Rapid Character Recognition,” *J. Ilmu Komput. dan Inf.*, vol. 18, no. 1, pp. 127–140, 2025, doi: 10.21609/jiki.v18i1.1443.
- [11] I. G. S. Mas Diyasa, P. A. Wijaya, and Y. V. Via, “Balinese Script Handwriting Recognition Using CNN and ELM Hybrid Algorithms,” *J. Nas. Pendidik. Tek. Inform.*, vol. 14, no. 1, pp. 49–59, 2025, doi: <https://doi.org/10.23887/janapati.v14i1.87524>.
- [12] A. Mulyanto, E. Susanti, F. Rossi, W. Wajiran, and R. I. Borman, “Penerapan Convolutional Neural Network (CNN) pada Pengenalan Aksara Lampung Berbasis Optical Character Recognition (OCR),” *JEPIN (Jurnal Edukasi dan Penelit. Inform.*, vol. 7, no. 1, pp. 52–57, 2021, doi: <https://doi.org/10.26418/jp.v7i1.44133>.
- [13] A. Rahim, F. Sthevanie, and K. N. Ramadhani, “Klasifikasi Aksara Lontara dari Sulawesi Selatan Menggunakan CNN,” *Log. J. Penelit. Inform.*, vol. 2, no. 2, pp. 35–42, 2024, doi: <https://doi.org/10.25124/logic.v2i2.8812>.
- [14] H. Guo, D. Yang, Y. Liu, and J. Zhao, “Script identification of ancient books by Chinese ethnic minorities using multi-branch DCNN and SPP,” *Pattern Anal. Appl.*, vol. 26, no. 2, pp. 809–821, 2023, doi: 10.1007/s10044-023-01146-y.
- [15] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, “GhostNet: More Features from Cheap Operations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1580–1589. doi: 10.1109/CVPR42600.2020.00165.
- [16] E. D. B. Sudewo, M. K. Biddinika, and A. Fadlil, “DenseNet Architecture for Efficient and Accurate Recognition of Javanese Script Hanacaraka Character,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 23, no. 2, pp. 453–464, 2024, doi: 10.30812/matrik.v23i2.xxx.
- [17] A. Prasetiadi, J. Saputra, I. Ramadhanti, A. D. Sripamuji, and R. R. Amalia, “Minimalist DCT-based Depthwise Separable Convolutional Neural Network Approach for Tangut Script,” *J. Dinda Data Sci. Inf. Technol. Data Anal.*, vol. 3, no. 2, pp. 59–64, Jul. 2023, doi: 10.20895/dinda.v3i2.1106.
- [18] M. A. Prameswari, M. D. Sulistiyo, and A. F. Ihsan, “Classification of Handwritten Sundanese Script via Transfer Learning on CNN-Based Architectures,” 2023, pp. 401–406. doi: 10.1109/ice3is59323.2023.10335382.
- [19] J. Ouyang, H. He, Y. He, and H. Tang, “Dog recognition in public places based on convolutional neural network,” *Int. J. Distrib. Sens. Networks*, vol. 15, no. 5, 2019, doi: 10.1177/1550147719829675.
- [20] Y. Zheng and Q. He, “Research and comparison of lightweight U-Net based on GhostNets for medical and remote sensing images,” in *2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 2023, pp. 2082–2086. doi: 10.1109/ICSP58490.2023.10248608.
-

- [21] D. R. K. Babu, P. R. Sowjanya, K. S. Trisha, and G. Kalyan, “TL-GhostNet: Transfer Learning Enhanced GhostNet for Attack Classification while Handling Big Data with Map-Reduce Framework,” in *2026 7th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, 2026, pp. 1444–1450. doi: 10.1109/ICMCSI67283.2026.11412844.
- [22] S. Kumar, A. Kumar, and K. Kumar, “GhostNet-YOLO Algorithm for Object Detection in UAV Image,” in *2023 Seventh International Conference on Image Information Processing (ICIIP)*, 2023, pp. 293–299. doi: 10.1109/ICIIP61524.2023.10537737.
- [23] Z. Wang *et al.*, “Micro-expression recognition method based on improved GhostNet combined with attention mechanism,” in *2025 5th International Conference on Big Data Engineering and Education (BDEE)*, 2025, pp. 119–123. doi: 10.1109/BDEE67464.2025.00025.