

Jaringan Syaraf Tiruan Dalam Pengenalan Pola Aksara Batak Simalungun Menggunakan Convolutional Neural Network

Muhammad Fadil¹, Hafizah², Azlan³

^{1,2,3} Sistem Informasi, STMIK Triguna Dharma

Email: ¹mf80177@gmail.com, ²hafizah22isnartiilyas@gmail.com, ³Azlansaja19@gmail.com

Email Penulis Korespondensi: mf80177@gmail.com

Abstrak

Aksara Batak Simalungun merupakan warisan budaya tak benda yang mulai terpinggirkan di era digital. Penelitian ini bertujuan mengembangkan sistem pengenalan pola aksara Batak Simalungun secara otomatis menggunakan algoritma Convolutional Neural Network (CNN). Dataset terdiri dari 570 citra tulisan tangan dari sepuluh responden, masing-masing menulis 19 huruf sebanyak tiga kali. Gambar diambil menggunakan kamera ponsel dan melalui tahapan pra-proses seperti resizing, konversi ke RGB dan normalisasi. Arsitektur CNN dirancang khusus dengan lapisan konvolusi, ReLU, pooling, Flatten dan fully connected layer. Hasil pengujian menunjukkan model mampu mengenali aksara dengan akurasi tinggi, yaitu sebesar 91,98%, serta nilai presisi, recall, dan f1-score yang baik. Penelitian ini menunjukkan bahwa CNN efektif dalam mengenali pola visual kompleks dan berpotensi digunakan untuk pelestarian budaya melalui aplikasi edukatif.

Kata Kunci: Aksara Batak Simalungun, Jaringan Syaraf Tiruan, Convolutional Neural Network, Pengenalan Pola, Aplikasi Edukatif.

Abstract

The Batak Simalungun script is an intangible cultural heritage that is increasingly marginalized in the digital era. This study aims to develop an automated recognition system for Batak Simalungun script patterns using the Convolutional Neural Network (CNN) algorithm. The dataset consists of 570 handwritten images collected from ten respondents, each writing 19 characters three times. The images were captured using a mobile phone camera and underwent several preprocessing steps, including resizing, RGB conversion, and normalization. The CNN architecture was specifically designed with convolutional layers, ReLU activations, pooling layers, flattening, and fully connected layers. Experimental results show that the model is capable of recognizing the characters with a high accuracy of 91.98%, along with good precision, recall, and F1-score values. This study demonstrates that CNN is effective in recognizing complex visual patterns and has the potential to be utilized for cultural preservation through educational applications.

Keywords: Script Batak Simalungun, Artificial Neural Network, Convolutional Neural Network, Pattern Recognition, educational application.

1. PENDAHULUAN

Aksara Batak Simalungun ialah aksara tradisional masyarakat Simalungun dan Batak di Sumatra Utara. Aksara Batak Simalungun merupakan salah satu warisan budaya Indonesia. Aksara Batak Simalungun sendiri mulai banyak dilupakan oleh generasi saat ini. Banyak hal yang menyebabkan Aksara Batak Simalungun ini dilupakan diantaranya adalah dengan gencarnya perkembangan teknologi yang menggunakan bahasa asing sehingga Aksara Batak Simalungun ini tidak mungkin untuk dipakai dalam perkembangan teknologi tersebut[1].

Aksara yang termasuk dalam warisan budaya tak benda berasal dari beberapa suku di Indonesia yang harus tetap di ajarkan kepada generasi berikutnya agar tidak hilang karena gencarnya perkembangan teknologi. Namun dalam hal tersebut ada berbagai macam rintangan salah satu nya adalah kurang pemanfaatan teknologi sebagai sarana pengajaran. Dengan perkembangannya yang cukup pesat teknologi menjadi salah satu solusi dalam sarana pengajaran.

Banyak cabang ilmu komputer yang dapat menyelesaikan berbagai macam masalah yang sangat kompleks. Hal tersebut dibuktikan dengan berbagai macam penelitian di bidang *datamining*, jaringan syaraf tiruan, sistem pendukung Keputusan dan banyak bidang lainnya[2]. Jaringan Saraf Tiruan (ANN) adalah model komputasi yang terinspirasi dari jaringan saraf biologis manusia. Mereka digunakan untuk memodelkan, mengenali pola, dan memahami data yang kompleks, serta telah menjadi dasar untuk banyak aplikasi kecerdasan buatan[3].

Algoritma *Convolutional Neural Network*, *Convolutional Neural Network* (CNN) adalah pengembangan dari *Multilayer Perceptron* (MLP) yang termasuk dalam *neural network* bertipe *feed forward* (bukan berulang). CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. CNN digunakan untuk menganalisis gambar visual, mendeteksi dan mengenali objek pada *image*, yang merupakan vektor berdimensi tinggi yang akan melibatkan banyak parameter untuk mencirikan jaringan[4].

Algoritma ini dipilih karena telah terbukti dalam penelitian sebelumnya mampu mencapai akurasi lebih dari 80% pada berbagai studi kasus lainnya. Berdasarkan keunggulan tersebut, penelitian ini diyakini mampu menghasilkan model dengan tingkat akurasi yang baik untuk pengenalan aksara Batak Simalungun. Beberapa penelitian telah berhasil menerapkan CNN untuk klasifikasi aksara Jawa. Sulaksana (2020) menggunakan CNN untuk mengidentifikasi aksara Jawa pada naskah kuno dengan akurasi 82,4%[5].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

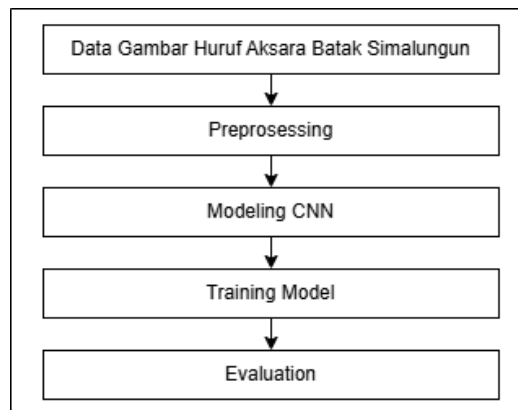
Dalam penelitian ini data yang digunakan yaitu citra tulisan Aksara Batak Simalungun yang berekstensi ‘.jpg’. Jumlah data yang di kumpulkan 570 citra Aksara Batak Simalungun yang diperoleh dari 10 responden yang di tulis ulang menggunakan alat tulis yang berbeda baik warna ataupun jenis, masing-masing menulis 19 huruf sebanyak 3 kali, menghasilkan total 57 huruf per responden. Data dikumpulkan dengan memotret setiap huruf secara individual menggunakan kamera *handphone*.

Tabel 1. Rancangan Blanko Kosong

Aksara Simalungun	Latin	Tulisan 1	Tulisan 2	Tulisan 3
	a			
	ha			
	ba			

2.2 Penerapan Algoritma Convolutional Neural Network

Algoritma *Convolutional Neural Network* (CNN) adalah metode pembelajaran terbimbing (*supervised learning*) yang dirancang untuk mengenali pola dalam data citra. CNN menggunakan serangkaian lapisan konvolusi, *pooling*, dan *fully connected* untuk mengekstrak fitur penting, mengurangi dimensi data, dan menghasilkan *output* berupa klasifikasi. Kerangka kerja algoritma CNN dapat dilihat pada gambar berikut.



Gambar 1. Kerangka Kerja Penerapan *Convolutional Neural Network*

Kerangka kerja tersebut dapat dijadikan sebagai gambaran dalam penggunaan *Convolutional Neural Network* untuk menyelesaikan permasalahan pengenalan pola huruf Aksara Batak Simalungun, berikut tahapan yang telah disusun.

1. Data Gambar Aksara Batak Simalungun
Data yang dimanfaatkan dalam penelitian ini citra dari gambar huruf Aksara Batak Simalungun sebanyak 570, yang masing-masing ditulis ulang oleh 10 responden.
2. Preprocessing
Pada tahapan *preprocessing* ini, gambar yang semula memiliki ukuran bervariasi, baik lebih kecil atau lebih besar dari 128x128 piksel, diubah menjadi seragam dengan ukuran tetap, yaitu 128x128 piksel.
3. Modeling CNN
Pada penerapan CNN untuk pengenalan pola aksara digunakan model *transfer learning* yaitu *MobileNetV2* yang setelah dilakukan percobaan akurasi tersebut menghasilkan akurasi terbaik. Untuk mengoptimalkan akurasi, dilakukan proses *trial and error* guna menentukan *hyperparameter* terbaik. Berdasarkan percobaan yang dilakukan, diperoleh nilai *hyperparameter* sebagaimana ditunjukkan pada Tabel berikut.

Tabel 2. *Hyperparameter* Add model *MobileNetV2*

No	<i>Hyperparameter</i>	
1	<i>Split data</i>	80%Train, 10%Valid, 10% Test
2	<i>Batch size</i>	32
3	<i>Layers</i>	6

4	<i>Epoch</i>	20
5	<i>Optimizer</i>	<i>Adam</i>
6	<i>Learning rate</i>	0.001

data dibagi menjadi 80% *Train*, 10% *Valid*, 10% *Test*, Dimana menggunakan *batch size* 32 atau sampel data yang disebarkan ke *Neural Network* sebanyak 32, yang berarti Proses pelatihan dilakukan oleh *Neural Network* hingga selesai, kemudian dilanjutkan dengan mengambil 32 sampel berikutnya dari 570 data, dan proses ini berulang. Sementara itu, jumlah 6 *layer* menunjukkan bahwa arsitektur terdiri atas enam lapisan. Selanjutnya, digunakan *optimizer Adam* untuk menyempurnakan bobot model secara bertahap berdasarkan data pelatihan. Pada tahap akhir, pelatihan menggunakan *learning rate* sebesar 0,001. Semakin kecil nilai *learning rate*, semakin teliti model dalam mempelajari data pelatihan.

a. *Input Layer*

Citra berukuran 128×128 piksel dengan 3 saluran warna (RGB) kemudian melalui proses prapengolahan dan dimasukkan ke dalam *input layer*. Nilai masukan dan keluaran dari *layer* ini berupa matriks berukuran 128×128 dengan rentang nilai [-1, 0, 1] serta saluran warna RGB.

b. Menambahkan Model

MobileNetV2 salah satu arsitektur *Deep Convolutional neural network* dengan mengadopsi total jumlah lapisan sejumlah 32 lapisan dan juga arsitektur ini didesain khusus untuk perangkat yang memiliki sumber daya latensi yang rendah seperti *smartphone* sehingga arsitektur ini dapat bekerja dengan efektif pada berbagai kondisi perangkat yang ada[6]. Model *MobileNetV2*, yang sebelumnya telah dilatih menggunakan metode *transfer learning*, dapat digunakan untuk mengolah dataset dengan *input* berupa gambar berukuran 128 piksel dan memiliki 3 saluran warna (*channel*). Keluaran (*output*) dari tahap konvolusi pertama akan menjadi masukan (*input*) untuk tahap konvolusi berikutnya secara berurutan. Dimensi gambar yang masuk adalah 128x128x3, dan hasil keluarannya pada tahap ini berukuran 4x4x1280, dengan jumlah total parameter sebanyak 2.257.984. Detailnya sebagai berikut

Model: "sequential_1"

Layer (type)	Output Shape	Param #
mobilenetv2_1.88_224 (Functional)	(None, 7, 7, 1280)	2257984
conv2d_1 (Conv2D)	(None, 7, 7, 32)	368672
max_pooling2d_1 (MaxPooling2D)	(None, 3, 3, 32)	0
dropout_1 (Dropout)	(None, 3, 3, 32)	0
flatten_1 (Flatten)	(None, 288)	0
dense_1 (Dense)	(None, 19)	5491

Total params: 2,632,147
 Trainable params: 174,163
 Non-trainable params: 2,257,984

Gambar 2. Output Tahapan Menambahkan *MobileNetV2*

c. *Convolutional Layer*

Layer konvolusi digunakan untuk mengekstraksi fitur-fitur penting dari citra *input*. Pada setiap *layer* konvolusi, terdapat filter yang digeser melintasi citra untuk melakukan operasi konvolusi. Operasi ini mengalikan bobot dalam filter dengan piksel yang diamati dan kemudian menjumlahkannya[7]. Pada tahap ini, data masukan yang diperoleh sebelumnya akan digunakan, yaitu *output image* dari *MobileNetV2* berukuran 4x4x1280. Proses konvolusi dengan 32 filter, kernel 3x3, *padding same* (0), fungsi aktivasi *ReLU*, dan *stride*= 1. Selanjutnya, proses konvolusi dilakukan dengan menerapkan filter pada nilai piksel yang telah disiapkan. Data piksel dari citra yang memiliki tiga *channel* (*Red*, *Green*, dan *Blue*) diambil masing-masing untuk di proses pada tahap konvolusi. Nilai piksel dari ketiga *channel* (*Red*, *Green*, *Blue*) akan dinormalisasi terlebih dahulu menggunakan metode *min-max normalization* untuk mempercepat proses pelatihan dan menjaga kestabilan jaringan. Normalisasi ini bertujuan untuk mengubah rentang nilai piksel menjadi [0, 1], sehingga lebih sesuai untuk pemrosesan dalam jaringan saraf konvolusional.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Digunakan *kernel* berukuran 3×3 dengan nilai seperti yang ditunjukkan pada matriks berikut:

$$K = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Operasi ini dilakukan dengan mengalikan elemen-elemen dari *kernel* terhadap elemen-elemen yang bersesuaian pada setiap jendela dari nilai *input*, kemudian menjumlahkan hasilnya untuk menghasilkan satu nilai keluaran.

$$O(i,j) = \sum_{i=1}^f \sum_{j=1}^f (I_{i,j} \cdot K_{i,j})$$

total hasil konvolusi dari ketiga *channel* yang telah di totalkan akan diproses menggunakan *ReLU* (*Rectified Linear Unit*). *ReLU* merupakan operasi untuk mengenalkan *non-linearitas* dan meningkatkan representasi dan model. Fungsi aktivasi *ReLU* akan menghilangkan *vanishing gradient* dengan cara menerapkan fungsi $f(x) = \max(0, x)$ atau aktivasi elemen akan dilakukan saat berada di ambang batas 0[8].

d. *Pooling Layer*

Output Dari tahap konvolusi akan digunakan sebagai masukan pada lapisan *pooling*. *Pooling Layer* adalah lapisan yang digunakan untuk menghasilkan sub-sampel peta fitur yang dihasilkan pada lapisan sebelumnya. Lapisan ini digunakan untuk mengetahui apakah suatu fitur tersedia atau tidak pada gambar yang dijadikan masukan. Ada beberapa metode *pooling* yang dapat digunakan pada *pooling layer*, beberapa diantaranya adalah *max pooling*, *min pooling*, *global average pooling* (GAP) dan *global max pooling*[9]. Pada proses ini diterapkan teknik *max pooling* dengan ukuran *kernel* 2x2 dan *stride* 2, menggunakan hasil konvolusi.

e. *Dropout Layer*

Dropout berfungsi untuk mengurangi kompleksitas model dan mencegah *overfitting* dengan mengabaikan sejumlah unit *neuron* secara acak selama proses pelatihan. Meskipun sejumlah *neuron* dinonaktifkan selama pelatihan, dimensi *output* dari tetap tidak berubah, karena *dropout* hanya mempengaruhi aktivasi.

f. *Flatten Layer*

Flatten adalah proses merubah dari bentuk keluaran citra matriks dua dimensi pada *pooling layer* menjadi satu kolom (sebuah vektor satu dimensi)[10].

$$Flatten(X) = [x_1, x_2, x_3, x_4], \quad O = h \times w$$

g. *Fully Connected Layer*

Lapisan *Fully-connected* atau lebih dikenal dengan *dense layer* adalah lapisan dimana semua *neuron* aktivitas dari lapisan sebelumnya terhubung semua dengan *neuron* di lapisan selanjutnya seperti halnya jaringan syaraf tiruan[11]. Hasil keluaran dari *flatten layer* akan diteruskan ke tahap akhir model CNN, yaitu *Dense layer* yang menggunakan fungsi aktivasi *softmax*. *Softmax* merupakan fungsi yang berfungsi untuk menghitung probabilitas dari citra dan mengklasifikasikannya. Fungsi ini bekerja dengan merubah vektor bilangan nyata menjadi bilangan yang jika dijumlahkan berjumlah 1. Masukan dari vektor bisa berupa bilangan positif lebih dari satu ataupun nol, *softmax* nantinya akan merubah masukan tersebut menjadi probabilitas diantara bilangan 0 sampai 1[12].

$$\begin{aligned} \sum_{i=1}^N I_i * v_{ij} &= J_i \\ \sum_{i=1}^N J_i * w_{ij} &= H_i \\ \sum_{i=1}^N H_i * x_{ij} &= O_i \\ s(o_i) &= \frac{e^{o_i}}{\sum_{j=1}^n e^{o_j}} \end{aligned}$$

3. *Training Model*

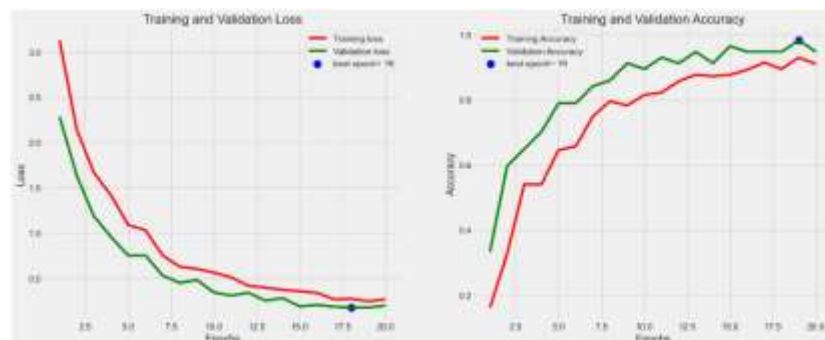
Proses pelatihan dilakukan sebanyak 20 *epoch* atau 20 kali iterasi, yang menghasilkan tingkat akurasi sebesar 91,98% dengan nilai *loss* dan *accuracy* sebagaimana ditampilkan pada Tabel berikut ini.

Tabel 3. *Training Model MobileNetV2*

No	Epoch	Loss		Accuracy	
		Loss	Val_Loss	Accuracy	Val_Accuracy
1	Epoch 1	3.1283	2.2865	0.1626	0.3333

No	Epoch	Loss		Accuracy	
		Loss	Val_Loss	Accuracy	Val_Accuracy
2	Epoch 2	2.1431	1.6358	0.3275	0.5965
3	Epoch 3	1.6691	1.1865	0.5407	0.6491
4	Epoch 4	1.4173	0.9588	0.5407	0.7018
5	Epoch 5	1.0919	0.7547	0.6462	0.7895
6	Epoch 6	1.0348	0.7566	0.6571	0.7895
7	Epoch 7	0.7559	0.5336	0.7495	0.8421
8	Epoch 8	0.6310	0.4522	0.7956	0.8596
9	Epoch 9	0.6052	0.4849	0.7824	0.9123
10	Epoch 10	0.5666	0.3467	0.8154	0.8947
11	Epoch 11	0.5123	0.3124	0.8220	0.9298
12	Epoch 12	0.4205	0.3434	0.8571	0.9123
13	Epoch 13	0.3998	0.2566	0.8769	0.9474
14	Epoch 14	0.3764	0.2844	0.8725	0.9123
15	Epoch 15	0.3600	0.1928	0.8769	0.9649
16	Epoch 16	0.3419	0.2089	0.8923	0.9474
17	Epoch 17	0.2719	0.1897	0.9143	0.9474
18	Epoch 18	0.2764	0.1798	0.8945	0.9474
19	Epoch 19	0.2493	0.1806	0.9297	0.9825
20	Epoch 20	0.2711	0.2010	0.9099	0.9474

Untuk melihat performa model dalam proses pelatihan dari model *MobileNetV2* di sertakan sebuah grafik yang memperlihatkan naik turun nya antar *training loss* dan *validation loss*.



Gambar 3. Grafik Hasil Training Model MobileNetV2

menggambarkan bagaimana model mempelajari data *training* secara bertahap dan mendalam. Hal ini terlihat dari pola kurva, di mana *training loss* dan *validation loss* secara perlahan menurun, yang menunjukkan seberapa baik performa model dalam mempelajari data untuk melakukan prediksi terhadap target. Sementara itu, nilai *training accuracy* dan *validation accuracy* secara bertahap meningkat, yang mengindikasikan bahwa akurasi model semakin membaik.

4. Evaluation

Hasil yang diperoleh nantinya digunakan sebagai dasar pertimbangan untuk menentukan apakah model memerlukan proses pelatihan ulang atau tidak, tergantung pada performa yang dicapai. *Confusion matrix* berfungsi untuk mengukur performa model atau tingkat akurasi dari suatu model klasifikasi terhadap masing-masing kelas yang diberikan[13].

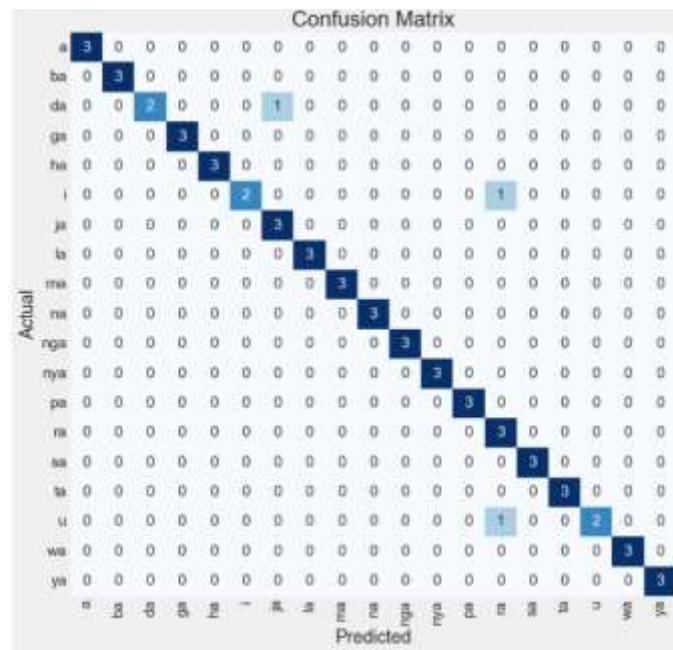
$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$Presisi\ Prediksi = \frac{TP}{TP + FP}$$

$$Sensitifitas\ Prediksi = \frac{TP}{TP + FN}$$

$$F1 - Score = 2 \times \frac{precision \times recall}{precision + recall}$$

Proses pelatihan menggunakan *MobileNetV2* dilakukan dengan 570 data dapat di evaluasi kembali untuk pengembangan selanjutnya yang di hasilkan oleh proses *confusion matrix*.



Gambar 4. *Confusion Matrix* Hasil Pelatihan Dengan *MobileNetV2*

3. HASIL DAN PEMBAHASAN

Sistem yang dikembangkan merupakan implementasi Jaringan Syaraf Tiruan untuk pengenalan pola Aksara Batak Simalungun menggunakan algoritma *Convolutional Neural Network*. Berikut ini disajikan hasil yang diperoleh dari sistem tersebut.

3.1 Tampilan Antarmuka Sistem

Gambar berikut menampilkan desain antarmuka (*interface*) dari sistem yang telah dikembangkan.

1. Halaman utama berfungsi sebagai pusat kontrol yang mengarahkan pengguna ke berbagai fitur utama, seperti halaman pelatihan jaringan, pengujian, dan riwayat pelatihan.



Gambar 5. Tampilan Halaman Utama

2. Halaman Pelatihan Dataset

Antarmuka pelatihan dataset berfungsi untuk melakukan proses pelatihan dataset, menghitung jumlah data yang digunakan, menampilkan hasil pemrosesan dataset yang telah dilatih, serta menghasilkan model yang akan digunakan dalam tahap pengenalan pola.

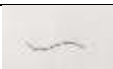
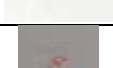
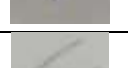
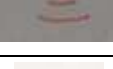
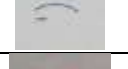
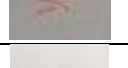


Gambar 8. Halaman Riwayat Pelatihan

3.2 Hasil Uji Sistem

Berikut ini adalah hasil pengujian sistem terhadap setiap huruf. Pada tahap ini menggunakan 19 sampel pada setiap huruf.

Tabel 4. Hasil Pengujian Sistem

Sampel Pola	Latin	Hasil Pengenalan	Keterangan	Sampel Pola	Latin	Hasil Pengenalan	Keterangan
	a	a	Valid		nga	nga	Valid
	ba	ba	Valid		nya	nya	Valid
	da	da	Valid		pa	pa	Valid
	ga	ga	Valid		ra	ra	Valid
	ha	ha	Valid		sa	sa	Valid
	i	i	Valid		ta	ta	Valid
	ja	ja	Valid		u	u	Valid
	la	la	Valid		wa	wa	Valid
	ma	ma	Valid		ya	ya	Valid
	na	na	Valid				

4. KESIMPULAN

Mampu mengenali pola huruf Aksara Batak Simalungun menggunakan algoritma *Convolutional Neural Network* (CNN) dengan tingkat akurasi 91,98. Proses ini melibatkan tahap pemrosesan citra, dilanjutkan dengan pelatihan model menggunakan CNN berdasarkan parameter yang telah ditentukan. Setelah pelatihan, sistem diuji menggunakan data yang telah dilatih sebelumnya untuk mengenali huruf-huruf Aksara Batak Simalungun. Namun disebabkan keterbatasan dalam jumlah dan variasi data menjadi faktor utama yang membatasi optimalisasi model.

UCAPAN TERIMA KASIH

Bersyukur yang sebesar-besarnya disampaikan kepada Allah SWT atas rahmat dan karunia-Nya, sehingga penulisan jurnal ini dapat diselesaikan dengan baik. Penulis juga menyampaikan apresiasi dan rasa hormat kepada Ibu Hafizah, S.Kom., M.Kom dan Bapak Azlan, S.Kom., M.Kom. atas bimbingan, arahan, serta ilmu yang telah diberikan selama

proses penyusunan jurnal ini. Tak lupa, penulis mengucapkan terima kasih kepada seluruh dosen dan staf STMIK Triguna Dharma atas segala bantuan, informasi, dan dukungan yang telah diberikan selama masa studi.

DAFTAR PUSTAKA

- [1] Y. F. Tarigan, B. H. Hayadi, and A. H. Nasyuha, "Implementasi Jaringan Saraf Tiruan Pengenalan Pola Aksara Batak Simalungun Menggunakan Kohonen Self Organizing Map," *Journal of Computer System and Informatics (JoSYC)*, vol. 3, no. 4, pp. 392–404, Sep. 2022, doi: 10.47065/josyc.v3i4.1991.
- [2] Rudiansyah, Dicky Nofriansyah, and Juniar Hutagalung, "24.+Rudiansyah,+Dicky+Nofriansyah,+Juniar+Hutagalung," *JURNAL SISTEM INFORMASI TGD*, vol. 3, pp. 1036–1043, Nov. 2024, Accessed: May 19, 2025. [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jsi>
- [3] R. Hartono and A. Zein, "PENERAPAN ALGORITMA GENETIKA DAN JARINGAN SYARAF TIRUAN DALAM PENJADWALAN MATA KULIAH Studi Kasus : Prodi Sistem Informasi Universitas Pamulang," *Jurnal Ilmu Komputer JIK*, vol. VI, no. 03, pp. 6–10, 2023.
- [4] P. Adi Nugroho, I. Fenriana, and R. Arijanto, "IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA EKSPRESI MANUSIA," *JURNAL ALGOR*, vol. 2, no. 1, 2020, [Online]. Available: <https://jurnal.buddhidharma.ac.id/index.php/algor/index>
- [5] N. Syafi'al Arief, W. Anggara Putra, D. Septhian, and R. Pratama, "Implementasi Cnn Arsitektur Mobilenetv2 Untuk Klasifikasi Tulisan Aksara Jawa."
- [6] S. A. Maulana *et al.*, "DETEKSI BURUNG MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN MODEL ARSITEKTUR MOBILENETV2," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 4, pp. 6108–6114, 2024.
- [7] D. R. R. Putra and R. A. Saputra, "IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK MENDETEKSI PENGGUNAAN MASKER PADA GAMBAR," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 11, no. 3, pp. 710–714, Aug. 2023, doi: 10.23960/jitet.v11i3.3286.
- [8] F. Paraijun *et al.*, "Implementasi Algoritma Convolutional Neural Network Dalam Mengklasifikasi Kesegaran Buah Berdasarkan Citra Buah," *KILAT*, vol. 11, no. 1, pp. 1–9, Apr. 2022, doi: 10.33322/kilat.v11i1.1458.
- [9] F. Ramadhani, A. Satria, and S. Salamah, "Implementasi Algoritma Convolutional Neural Network dalam Mengidentifikasi Dini Penyakit pada Mata Katarak," *sudo Jurnal Teknik Informatika*, vol. 2, no. 4, pp. 167–175, Dec. 2023, doi: 10.56211/sudo.v2i4.408.
- [10] H. Herdianto and D. Nasution, "IMPLEMENTASI METODE CNN UNTUK KLASIFIKASI OBJEK," *METHOMIKA Jurnal Manajemen Informatika dan Komputerisasi Akuntansi*, vol. 7, no. 1, pp. 54–60, Apr. 2023, doi: 10.46880/jmika.Vol7No1.pp54-60.
- [11] K. Azmi, S. Defit, and U. Putra Indonesia YPTK Padang Jl Raya Lubuk Begalung-Padang-Sumatera Barat, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," *Jurnal Unitek*, vol. 16, no. 1, pp. 28–40, 2023.
- [12] W. Wijaya Kusuma, R. Rizal Isnanto, A. Fauzi, and P. Korespondensi, "DenseNet121 Menggunakan Kerangka Kerja TensorFlow untuk Deteksi Jenis Hewan," *Jurnal Teknik Komputer*, vol. 1, no. 4, pp. 141–147, 2023, doi: 10.14710/jtk.v1i4.37009.
- [13] A. Rizal, I. 1✉, and T. N. Suharsono, "Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Citra Jamur Berbasis Mobile," *INNOVATIVE: Journal Of Social Science Research*, vol. 3, pp. 864–875, 2023.