

Implementasi *Convolutional Neural Network* (CNN) Untuk Pengenalan Tulisan Tangan Aksara Bima

Ahmad Angga Handoko¹
Mochamad Alfian Rosid^{2*}
Uce Indahyanti³

^{1,2,3}Informatika, Universitas Muhammadiyah Sidoarjo, Jl. Mojopahit No.666 B, Sidowayah, Celep, Kec. Sidoarjo, Kabupaten Sidoarjo, Jawa Timur, 61215 Indonesia

¹ahmadangga.h@protonmail.com, ²alfanrosid@umsida.ac.id, ³uceindahyanti@umsida.ac.id

*Penulis Korespondensi:

Mochamad Alfian Rosid
alfanrosid@umsida.ac.id

Abstrak

Indonesia dengan kaya akan ragam budaya yang dimilikinya, tentu mengakibatkan keberadaan beragam bahasa di dalamnya. Pada dialek suatu wilayah, terdapat lambang-lambang huruf yang mewakili pengungkapan bahasa khas daerah tersebut. Salah satu contoh bahasa yang memperlihatkan karakteristik unik dalam sistem penulisannya adalah bahasa Bima, yang dikenal dengan sebutan aksara Bima. Aksara Bima, atau yang sering disebut sebagai aksara Mbojo adalah sistem tulisan yang secara tradisional digunakan di wilayah Bima, terletak di Provinsi Nusa Tenggara Barat. Aksara Bima masih belum tersebar luas di kalangan masyarakat umum, sehingga penting untuk melestarikannya sebagai bagian dari warisan budaya suku Mbojo. Penelitian ini bertujuan untuk melatih komputer untuk mengenali karakter aksara Bima. Metode *Convolutional Neural Network* (CNN) digunakan dalam penelitian ini untuk melakukan pengenalan terhadap tulisan tangan aksara Bima. Dataset yang digunakan terdiri dari 2640 citra tulisan tangan aksara Bima dengan 22 kelas. Hasil penelitian menunjukkan kinerja model CNN yang handal, dengan akurasi 97,34%, presisi 97,56%, recall 97,34%, dan f1-score 97,31% pada data uji.

Kata Kunci: Aksara Bima; *Convolutional Neural Network*; Deep learning; Pengenalan; Tulisan Tangan

Abstract

Indonesia with its rich cultural diversity, certainly results in the existence of various languages in it. In the dialect of a region, there are letter symbols that represent the expression of the region's unique language. One example of a language that shows unique characteristics in its writing system is Bima, known as the Bima script. Bima script, or often referred to as Mbojo script, is a writing system traditionally used in the Bima region, located in West Nusa Tenggara Province. Bima script is still not widespread among the public, so it is important to preserve it as part of the cultural heritage of the Mbojo tribe. This research aims to train a computer to recognize Bima characters. The *Convolutional Neural Network* (CNN) method is used in this research to recognize Bima script handwriting. The dataset used consists of 2640 images of Bima script handwriting with 22 classes. The results showed a reliable performance of the CNN model, with an accuracy of 97,34%, precision 97,56%, recall 97,34%, and f1-score 97,31% on the test data.

Keywords: Bima Script; *Convolutional Neural Network*; Deep learning; Handwriting; Recognition

1. Pendahuluan

Indonesia sebagai negara dengan wilayah kepulauan terluas di dunia, terdiri dari 17.508 pulau, dihuni oleh berbagai suku bangsa yang menciptakan kekayaan dalam bentuk keragaman budaya dan tradisi [1]. Aksara Bima sebagai salah satu bentuk budaya, yang juga dikenal sebagai Aksara Mbojo digunakan di Kabupaten Bima, Nusa Tenggara Barat. Terdapat dua versi aksara Bima yaitu aksara Bima kuno dan aksara Bima/mbojo yang memiliki bentuk mirip dengan aksara lontara atau bugis. Aksara Bima dipakai untuk menulis bahasa Bima (Nggahi Mbojo) yang digunakan di bagian timur pulau Aksara Bima merupakan sistem penulisan yang digunakan untuk menulis

bahasa Bima (Nggahi Mbojo), yang secara linguistik digunakan di bagian timur pulau Sumbawa [2].

Aksara Bima sudah dikenal sejak berabad-abad yang lalu. Menurut catatan sejarah, kerajaan Bima telah menggunakan Aksara Mbojo (Aksara Bima) dalam aktivitas menulis naskah-naskah kerajaan sejak abad ke XIV M. Menurut Hj. Siti Maryam R. Salahuddin, Aksara Mbojo memiliki dua bentuk huruf. Pertama, terdiri dari satu huruf, mulai dari A hingga Y, dengan pengecualian Q, V, X, dan Z yang tidak memiliki bentuk hurufnya. Dalam beberapa kasus, jika terdapat huruf Q, digantikan dengan huruf K, misalnya kata "Qosdun" menjadi "Kosdun." Demikian pula, huruf Z diganti dengan J, seperti "zaman" menjadi "jaman" dan "zat" menjadi "jat." Mengenai huruf V dan X, masih perlu penelitian lebih lanjut untuk menentukan apakah mereka memiliki kata dalam bahasa Bima. Aksara Mbojo juga mirip dengan huruf hijaiyah dan memiliki tanda baca yang membedakan antara a, i, u, e, dan o [3].

Aksara Bima digunakan di tingkat pendidikan dasar dan menengah pertama yang termuat dalam peraturan Walikota Bima nomor 50 tahun 2019 [4]. Setelah menyelesaikan jenjang studi sekolah menengah pertama, tidak ada lagi pelajaran aksara Bima pada jenjang sekolah berikutnya. Dengan berkurangnya penggunaan aksara Mbojo pada saat ini ditakutkan dapat menyebabkan hilangnya kebudayaan yang diwariskan oleh Suku Mbojo [5]. Oleh karena itu penting untuk dilakukan edukasi tentang bentuk dan tulisan aksara Mbojo kepada masyarakat agar dapat mempertahankan kebudayaan yang telah diwariskan oleh suku Mbojo [6].

Pengenalan tulisan tangan aksara Bima dilakukan melalui pendekatan *deep learning* menggunakan algoritma *Convolutional Neural Network* (CNN) sebagai bagian dari upaya pelestarian aksara. Metode *Neural Network* merupakan salah satu pendekatan pembelajaran terawasi yang sudah memperoleh popularitas yang cepat dalam beberapa tahun terakhir karena kinerja yang baik [7]. Salah satu faktor yang mempengaruhi performansi atau nilai presisi yang dihasilkan adalah banyaknya iterasi. *Convolutional Neural Network* (CNN) adalah evolusi dari *Multi-Layer Perceptron* (MLP) yang dirancang untuk mengelola data dua dimensi, dan karena kedalaman jaringan yang dalam dan banyak diterapkan pada data gambar, CNN diklasifikasikan sebagai jenis jaringan saraf tiruan [8]. MLP untuk klasifikasi citra ini tidak cocok karena tidak mencatat informasi khusus dari citra dan setiap piksel dapat diinterpretasikan sebagai fitur yang bersifat independen jadi mendapatkan hasil yang tidak memuaskan [9].

Penelitian mengenai pengenalan tulisan tangan menggunakan kecerdasan buatan menjadi topik yang menarik bagi para peneliti dalam beberapa tahun terakhir. Salah satunya penelitian yang berjudul "Pengenalan Tulisan Tangan Karakter Aksara Batak Toba dengan Metode *Convolutional Neural Network* (CNN)" [10]. Tujuan dari penelitian ini adalah mengembangkan suatu model yang memiliki kemampuan untuk mengenali tulisan dalam aksara Batak Toba menggunakan *Convolutional Neural Network*. Hasil dari penelitian ini menunjukkan kinerja optimal pada pengujian pertama dengan tingkat akurasi mencapai 89.47%.

Pada penelitian selanjutnya yang berjudul "Perbandingan Arsitektur VGG16 Dan Resnet50 Untuk Rekognisi Tulisan Tangan Aksara Lampung" [11]. Penelitian ini bertujuan mengklasifikasikan citra tulisan tangan aksara lampung dengan membandingkan kinerja model deep learning VGG16 dan ResNet50. Hasil uji coba menunjukkan bahwa VGG16 mencapai tingkat akurasi sebesar 91% dan juga memberikan efisiensi waktu pelatihan yang lebih baik. Di sisi lain, ResNet50 mendapatkan akurasi 65% yang memerlukan waktu komputasi yang lebih lama dan memberikan tingkat akurasi yang lebih rendah.

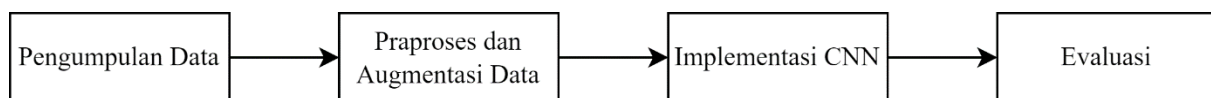
Pada penelitian lain dengan judul "Identifikasi Tulisan Tangan Aksara Jepang Hiragana Menggunakan Metode CNN Arsitektur VGG-16" [12]. Penelitian ini bertujuan mengidentifikasi objek tulisan tangan huruf Jepang Hiragana dengan metode CNN dengan arsitektur VGG-16. Dalam penelitian ini, tiga skenario berbeda digunakan: *optimizer* Adam, *optimizer* SGD, dan

optimizer RMSprop. Hasil penelitian terbaik di skenario *optimizer* Adam dengan nilai *accuracy* sebesar 97.6%. Penelitian yang berjudul “Vietnamese Handwritten Character Recognition Using Convolutional Neural Network”, bertujuan untuk membuat model yang efektif untuk mengenali karakter tulisan tangan Vietnam dengan menggunakan *Convolutional Neural Network* (CNN), yang terdiri dari tiga *convolutional layers* dan dua *fully connected layers*. Teknik *dropout* digunakan untuk mencegah terjadinya *overfitting*, menghasilkan tingkat akurasi 97% [13]. Imran Khandokar pada penelitian yang berjudul “Handwritten Character Recognition Using Convolutional Neural Network”. Bertujuan untuk menerapkan *Convolutional Neural Network* dalam pengenalan tulisan tangan menggunakan dataset NIST. Hasil pengujian menunjukkan bahwa akurasi sebesar 92.91% diperoleh dari 200 gambar *testing* dengan 1000 gambar *training* [14].

Pada penelitian ini menggunakan arsitektur CNN yang terdiri dari 4 *convolutional layers* dan 2 *fully connected layers*. Penerapan teknik *dropout* untuk mencegah model mengalami *overfitting* dan augmentasi data untuk memperbanyak variasi data dengan membuat data baru dari data yang sudah ada. Selain itu, melakukan *tuning* pada *hyperparameter* untuk memperoleh hasil akurasi yang optimal dengan 4 macam skenario. Tujuan dari penelitian ini untuk membangun model *Convolutional Neural Network* (CNN) yang mampu mengenali tulisan tangan aksara Bima.

2. Metode Penelitian

Tahapan penelitian ini terdiri dari pengumpulan data, praproses dan augmentasi data, implementasi CNN, dan evaluasi [15].

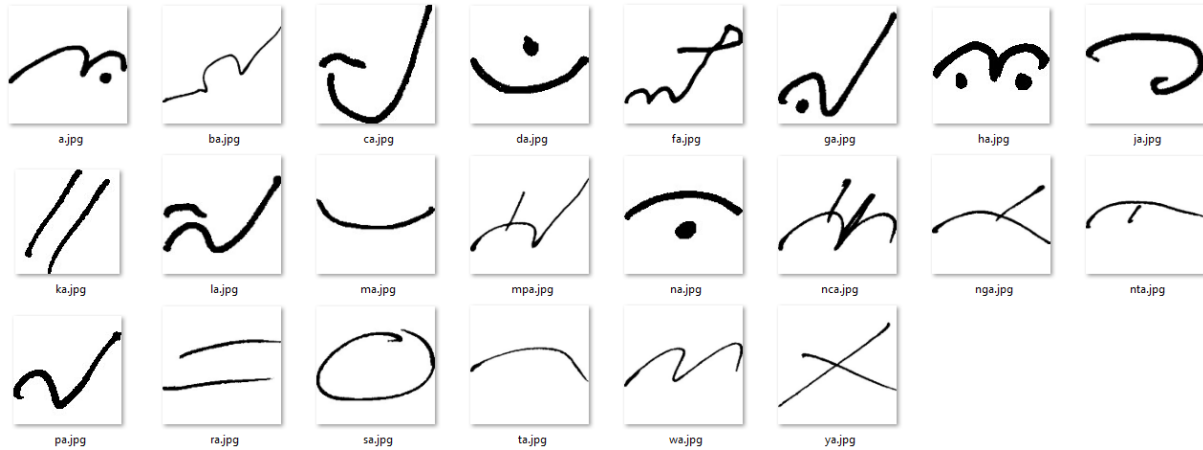


Gambar 1. Tahapan Penelitian

Pengumpulan Data

Data diperoleh dari penelitian dengan judul “Pengenalan Pola Tulisan Tangan Aksara Bima menggunakan Ciri Tekstur dan KNN” [5]. Proses pengumpulan data aksara Bima melibatkan sebanyak 10 orang responden. Responden diharapkan untuk menuliskan 22 karakter dalam aksara Bima, dengan masing-masing karakter ditulis sebanyak 12 kali. Setiap karakter akan dituliskan pada media kertas F4, menggunakan bolpoin berukuran 1.0 mm. Setelah itu, data akan dijadikan digital menggunakan *scanner* dengan pengaturan sebesar 600 dpi.

Data yang terkumpul dilakukan *cropping* sehingga menghasilkan 12 buah tulisan tangan untuk setiap karakternya. Setiap citra karakter tulisan aksara Bima dibedakan dengan cara diberi label sesuai dengan karakter masing-masing. Sehingga data yang didapat sebanyak 2640 citra. Hasil pengumpulan citra tulisan tangan aksara Bima dapat dilihat pada Gambar 2.



Gambar 2. Citra Tulisan Tangan Aksara Bima

Dataset terdiri dari 22 kelas yaitu a, ba, ca, da, fa, ga, ha, ja, ka, la, ma, mpa, na, nca, nga, nta, ra, pa, sa, ta, wa, dan ya. Data dibagi menjadi 80% untuk data latih dan 20% untuk data uji menghasilkan data latih sebanyak 2.112 dan data uji sebanyak 528 citra.

Praproses Dan Augmentasi Data

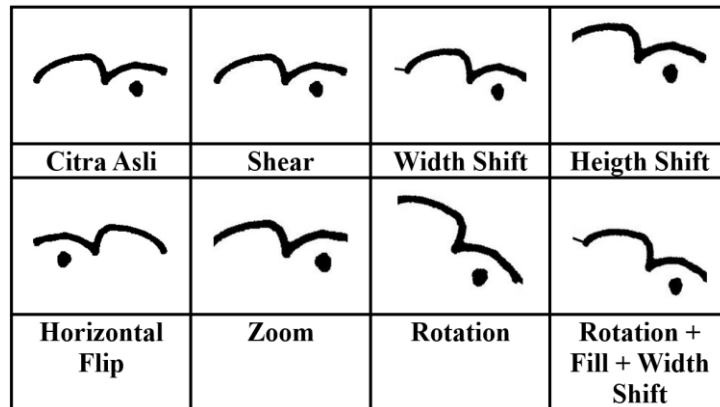
Praproses data dilakukan dengan merubah ukuran semua citra menjadi 180 x 180 piksel. Citra juga dikonversi menjadi 3 channel warna yaitu *Red, Green, and Blue* (RGB) serta melakukan *rescaling* dengan nilai $1./255$ seperti pada persamaan (1), agar rentang nilai RGB yang semula 0-255 menjadi 0 dan 1 [16][17]. Hal tersebut dilakukan supaya mempercepat proses *training* dan membutuhkan *memory* yang lebih sedikit [18].

$$output = input \times \frac{1}{255} \quad (1)$$

Augmentasi data adalah metode yang dapat digunakan untuk meningkatkan jumlah citra dalam dataset dengan melakukan transformasi pada citra asli. Dengan menggunakan augmentasi data dapat meningkatkan akurasi dari model, terutama untuk beberapa kelas yang salah diklasifikasi [19]. Transformasi umum yang biasanya digunakan adalah *flipping, rotating, scaling, and cropping* [20]. Adapun beberapa transformasi yang dilakukan dalam penelitian ini yaitu:

1. *Rotation* untuk merotasi citra secara acak.
2. *Shear* untuk mencukur citra secara acak.
3. *Width Shift* dan *Height Shift* untuk menggeser citra secara acak baik lebar maupun tinggi.
4. *Zoom* untuk memperbesar bagian citra secara acak.
5. *Horizontal Flip* untuk membalik citra secara horizontal.
6. *Filling* untuk mengisi piksel baru yang terbuat saat melakukan rotasi atau pergeseran secara lebar maupun tinggi.

Hasil praproses dan augmentasi data dapat dilihat pada Gambar 3.

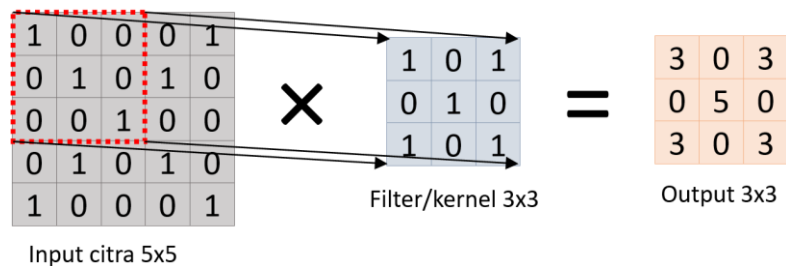


Gambar 3. Hasil Praproses Dan Augmentasi Data

Pembangunan Model CNN

Convolutional Neural Network adalah kombinasi dari algoritma *deep learning* dan jaringan syaraf buatan yang biasanya digunakan untuk memproses data piksel dan citra [21], [22]. Arsitektur CNN memiliki beberapa komponen yaitu: *Convolution layer*, *Rectified Linear Unit (ReLU)*, *Pooling layer*, *Fully connected layer*, dan *softmax activation*.

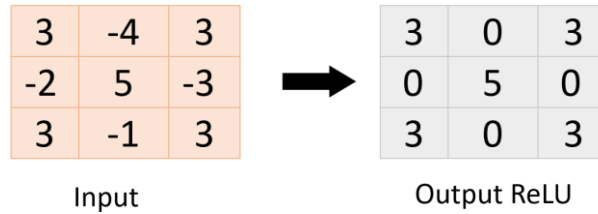
Convolution layer merupakan bagian penting dari CNN yang berfungsi untuk mengekstrak fitur pada citra [23]. Pada *layer* ini, terjadi proses konvolusi di mana filter diterapkan pada citra. Filter yang diterapkan biasanya berbentuk matriks dengan ukuran seperti 1x1, 3x3, atau 5x5 [24]. Proses konvolusi ini menghasilkan *feature map* yang kemudian akan dipergunakan pada *activation layer* [25]. Pada Gambar 4 menunjukkan proses *dot operation* dari citra masukan 5x5 dengan filter 3x3, setelah proses konvolusi selesai dibagian pertama, kemudian, filter akan bergeser secara bertahap hingga semua bagian dari citra telah diproses. Proses ini akan membentuk output berupa citra dengan ukuran 3x3.



Gambar 4. Convolutional Layer

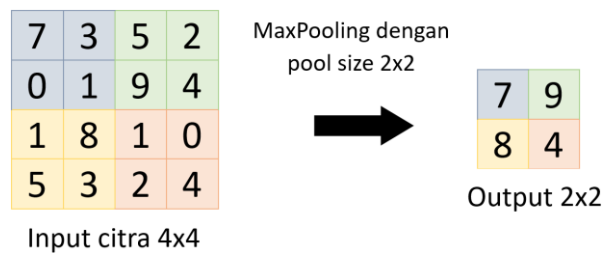
Rectified Linear Unit (ReLU) merupakan salah satu jenis *activation function* untuk memperkenalkan non-linearitas pada model CNN. Pada Gambar 5 adalah ilustrasi dari operasi ReLU, jika inputan x adalah positif maka keluaranya x , jika sebaliknya maka 0. Persamaan (2) merupakan rumus dari ReLU [21].

$$f(x) = \max(0, x) \quad (2)$$



Gambar 5. ReLU Function

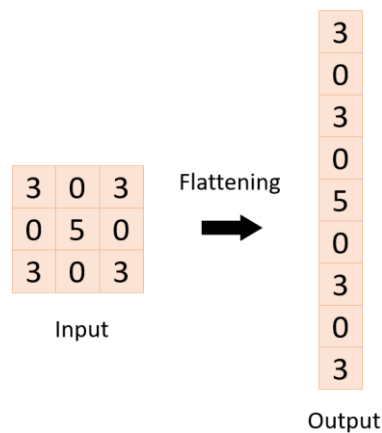
Pooling layer berfungsi untuk mengurangi dimensi citra namun tetap menjaga informasi penting yang terdapat dalam citra tersebut. Proses pooling dapat mengurangi sumber daya komputasi yang diperlukan untuk pemrosesan citra [26]. Pooling layer memiliki beberapa kategori yaitu MaxPooling, AveragePooling, GlobalMaxPooling, dan GlobalAveragePooling. MaxPooling digunakan dalam penelitian ini dengan size bawaan 2 x 2 yang akan memberikan keluaran berupa nilai terbesar dari citra. Pada Gambar 6 menunjukkan proses pooling.



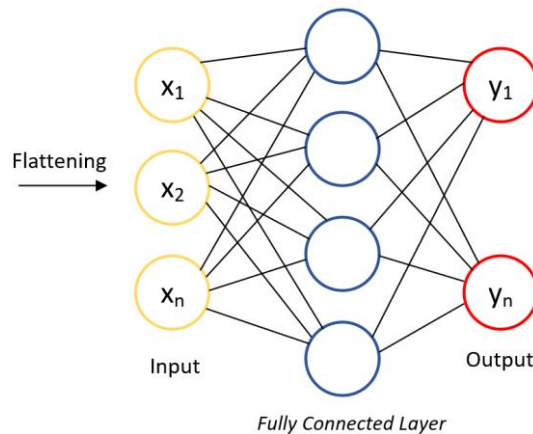
Gambar 6. Pooling Layer

Fully connected layer merupakan sebuah layer yang berfungsi untuk melakukan transformasi dimensi data agar dapat diklasifikasikan secara linier [27]. Sebelum memasuki fully connected layer setiap neuron pada convolutional layer perlu dikonversikan menjadi vektor, proses ini disebut flattening, diilustrasikan seperti pada Gambar 7. Setelah proses flattening setiap aktivitas dari layer sebelumnya dapat dihubungkan ke semua neuron dari fully connected layer seperti pada Gambar 8.

Fully connected layer menggunakan activation function softmax yang berfungsi untuk menentukan klasifikasi citra inputan [28]. Softmax mengeluarkan keluaran yang terletak dalam interval nilai antara 0 dan 1, dengan total probabilitas semua kemungkinan adalah 1. Teknik regularisasi dropout diterapkan sebelum fully connected layer untuk mengatasi potensi overfitting dan meningkatkan kecepatan pembelajaran.



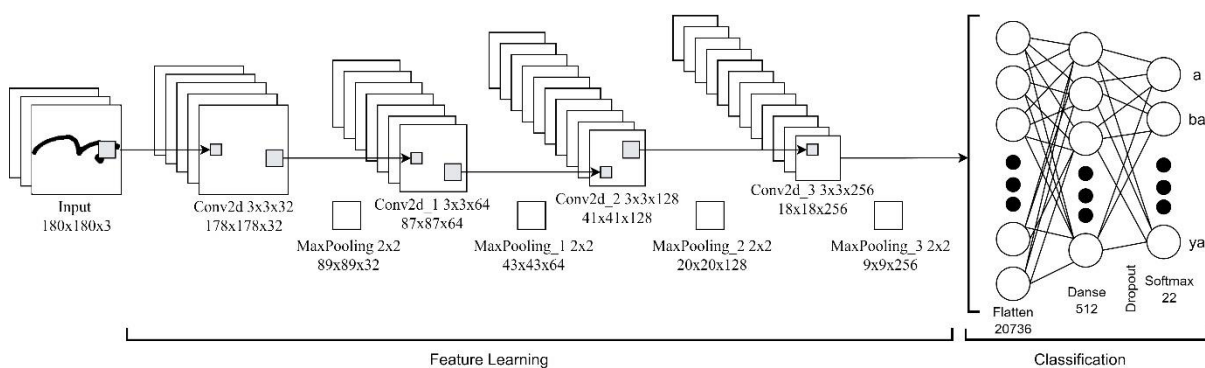
Gambar 7. Flattening



Gambar 8. Fully Connected Layer

Arsitektur CNN pada penelitian ini diilustrasikan pada Gambar 9 yang memiliki 11.017.046 parameter. Data tulisan tangan aksara Bima yang sudah melalui tahapan praproses dan augmentasi data dengan *shape* citranya adalah 180x180x3 kemudian masuk ke *convolution layer* pertama dengan filter 3x3 menghasilkan 178x178x32. Hasil tersebut akan dimasukkan ke *pooling layer* dengan *pool size* 2x2 dan *stride* bernilai 1 sehingga memberikan keluaran 89x89x32, proses tersebut berulang seterusnya sampai *convolutional layer* terakhir menjadi 9x9x256 sebelum masuk ke *fully connected layer*. *Convolutional layer* mengecilkan dimensi citra yang awalnya 180x180 menjadi 9x9 dan memperbesar *channel* yang awalnya 3 menjadi 256. Aktivasi ReLU digunakan pada setiap *convolutional layer* untuk mengurangi kemungkinan penurunan performa arsitektur karena adanya perhitungan negatif.

Layer terakhir yaitu *fully connected layer* untuk mengklasifikasi berdasarkan fitur yang sudah dipelajari di *convolutional layer*. Proses *flatten* menghasilkan 20.736 parameter. Pada *hidden layer* memiliki 512 *neurons* dengan aktivasi ReLU kemudian dilakukan *dropout*. Pada *output layer* dengan 22 *neurons* yang sesuai dengan jumlah kelas prediksi aksara Bima. Lapisan ini menggunakan fungsi aktivasi *softmax* untuk menghasilkan distribusi probabilitas dari prediksi kelas.



Gambar 9. Arsitektur CNN

Dalam proses *training* terdapat beberapa parameter yaitu step per *epoch* sebanyak data uji dibagi *batch size*, nilai *epoch*, *loss function* menggunakan *Sparse Categorical Cross Entropy (SCCE)*, *modelcheckpoint* dengan *save_best_only*, metrics berupa *accuracy*, *validation_data* menggunakan data uji untuk mengevaluasi *validation loss* dan *validation accuracy* pada setiap akhir *epoch*.

Evaluasi

Confusion matrix digunakan dalam proses evaluasi untuk mengetahui perbandingan hasil prediksi yang dilakukan model dengan label sebenarnya dari 22 kelas. *Confusion matrix* berfungsi untuk mengukur berbagai *metrics* yang berbeda dalam mengevaluasi kinerja model yang telah dikembangkan [29], [30], [31]. Beberapa *metrics* yang umum digunakan mencakup *accuracy*, *precision*, *recall*, dan *f1-score* [32][33]. Terdapat 4 kemungkinan yang terjadi seperti terlihat pada Gambar 10.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 10. Confusion Matrix [34].

1. *True Positive* (TP) : kondisi dimana aksara Bima diprediksi dengan benar sebagai kelas positif (*Positive*), dan memang benar-benar termasuk kelas yang sesuai (*True*).
2. *True Negative* (TN) : kondisi dimana aksara Bima diprediksi dengan kelas negatif (*Negative*), dan memang tidak termasuk kelas yang sesuai (*True*).
3. *False Positive* (FP) : kondisi dimana aksara Bima diprediksi dengan benar sebagai kelas positif (*Positive*), tetapi ternyata tidak termasuk dalam kelas yang benar (*False*).
4. *False Negative* (FN) : kondisi dimana aksara Bima diprediksi sebagai kelas negatif (*Negative*), tetapi ternyata termasuk dalam kelas yang benar (*True*).

Accuracy adalah perbandingan antara prediksi yang benar dengan seluruh data yang ada. Akurasi menjelaskan “Berapa persen aksara Bima yang diprediksi sesuai kelasnya dari keseluruhan data”.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (3)$$

Precision adalah perbandingan antara prediksi yang tepat positif dengan total prediksi yang dianggap positif. *Precision* menjelaskan “Berapa persen aksara Bima yang memang sesuai kelasnya dari keseluruhan aksara Bima yang diprediksi sesuai kelasnya”.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

Recall adalah perbandingan antara prediksi yang tepat positif dengan keseluruhan data yang memang benar. *Recall* menjelaskan “Berapa persen aksara Bima yang diprediksi sesuai kelasnya dibandingkan keseluruhan aksara Bima yang memang sesuai kelasnya”.

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

F1 Score merupakan *harmonic mean* dari *recall* dan *precision*.

$$F1\ Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (6)$$

3. Hasil

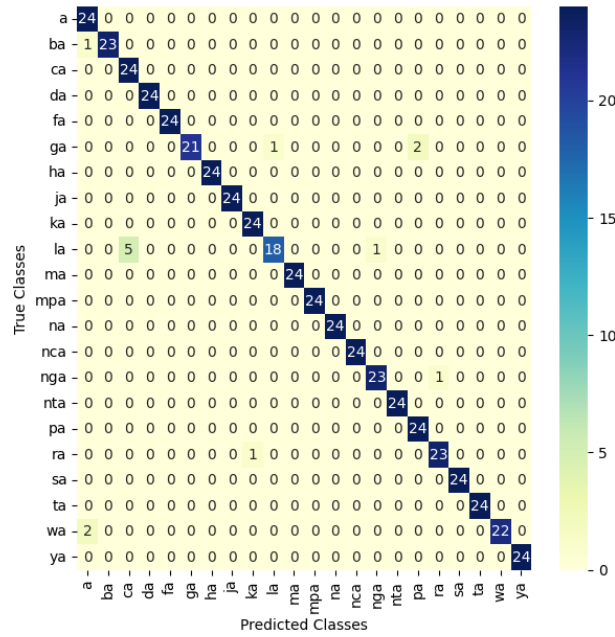
Penelitian ini dilakukan menggunakan *Google Colab* dengan bahasa pemrograman *Python* dan *library* seperti *Keras*, *NumPy*, *Matplotlib*, *Pandas*, dan *Scikit-learn*. Peneliti melakukan beberapa skenario untuk mengoptimalkan model dalam mengenali tulisan tangan aksara Bima. Terdapat 4 skenario, skenario pertama percobaan terhadap variasi nilai *epoch*, skenario kedua percobaan terhadap variasi nilai *learning rate*, skenario ketiga percobaan terhadap variasi jenis *optimizer*, dan skenario keempat percobaan terhadap variasi nilai *batch size*.

Tabel 1. Classification Report

No	Aksara	Precision	Recall	F1-score
1.	a	0.89	1.00	0.94
2.	ba	1.00	0.96	0.98
3.	ca	0.83	1.00	0.91
4.	da	1.00	1.00	1.00
5.	fa	1.00	1.00	1.00
6.	ga	1.00	0.88	0.93
7.	ha	1.00	1.00	1.00
8.	ja	1.00	1.00	1.00
9.	ka	0.96	1.00	0.98
10.	la	0.95	0.75	0.84
11.	ma	1.00	1.00	1.00
12.	mpa	1.00	1.00	1.00
13.	na	1.00	1.00	1.00
14.	nca	1.00	1.00	1.00
15.	nga	0.96	0.96	0.96
16.	nta	1.00	1.00	1.00
17.	pa	0.92	1.00	0.96
18.	ra	0.96	0.96	0.96
19.	sa	1.00	1.00	1.00
20.	ta	1.00	1.00	1.00
21.	wa	1.00	0.92	0.96
22.	ya	1.00	1.00	1.00
Akurasi pelatihan			94.36	
Akurasi validasi			97.34	
Macro AVG		97.56	97.34	97.31
Weighted AVG		97.56	97.34	97.31

Tabel 1 menunjukkan hasil percobaan terbaik dengan kombinasi *hyperparameter* dengan nilai *epoch* 100, *batch size* 32, *optimizer* Adam, dan *learning rate* 0,001 yaitu akurasi validasi mencapai 97,34% dari keseluruhan data uji sedangkan akurasi pelatihan sebesar 94,36% dari keseluruhan data latih. Terdapat beberapa kekurangan seperti nilai *precision* terendah pada aksara ke-3 (ca) sebesar 83%, nilai *recall* dan *f1-score* terendah pada aksara ke-10 (la) sebesar 75% dan 84%.

Confusion Matrix pada Gambar 11 menunjukkan beberapa kesalahan prediksi aksara Bima seperti aksara ke-2 (ba) diprediksi a, aksara ke-6 (ga) diprediksi la dan pa, aksara ke-15 (nga) diprediksi ra, aksara ke-18 (ra) diprediksi ka, aksara ke-21 (wa) diprediksi a, dan kesalahan prediksi terbanyak pada aksara ke-10 (la) diprediksi la dan nga.



Gambar 11. Confusion Matrix

4. Pembahasan

Percobaan Variasi Nilai Epoch

Skenario pertama dilakukan percobaan dengan menggunakan berbagai nilai *epoch* pada saat *training* model untuk mengetahui nilai *epoch* yang memberikan performa terbaik. Nilai *epoch* yang digunakan dalam percobaan ini yaitu 25, 50, 75, dan 100. Terdapat parameter lain yang digunakan saat proses percobaan seperti *optimizer* Adam dengan *learning rate* 0.001, *batch size* sebesar 32. Hasil percobaan disajikan dalam Tabel 2.

Tabel 2. Hasil Perbandingan Nilai Epoch

Epoch	Akurasi Training	Akurasi Validation	Loss Validation
25	91,80	94,69	0,1696
50	89,15	96,59	0,1158
75	94,31	96,78	0,1022
100	94,36	97,34	0,0799

Berdasarkan Tabel 2, bertambahnya nilai *epoch* yang digunakan dapat meningkatkan performa dari model CNN dilihat dari akurasi *validation* yang meningkat dan menurunnya nilai *loss validation*. Performa terbaik pada *epoch* dengan nilai 100 memiliki akurasi *validation* sebesar 97,34% yang lebih tinggi dari penggunaan nilai *epoch* lainnya. Berdasarkan hasil tersebut, *epoch* 100 merupakan nilai yang optimal untuk pengenalan tulisan tangan aksara Bima.

Percobaan Variasi Nilai Learning Rate

Skenario kedua melakukan percobaan dengan menggunakan berbagai nilai *learning rate* dari $0.1(10^{-1})$ sampai $0.00001(10^{-5})$ untuk mengetahui nilai *learning rate* yang memberikan performa terbaik. Pada percobaan ini nilai *epoch* yang digunakan adalah 100, *batch size* dengan nilai 32 dan *optimizer* menggunakan Adam. Hasil percobaan dapat dilihat pada Tabel 3.

Tabel 3. Hasil Perbandingan Nilai Learning Rate

Learning Rate	Akurasi Training	Akurasi Validation	Loss Validation
---------------	------------------	--------------------	-----------------

0,1	4,54	4,54	3,1031
0,01	4,54	4,54	3,0912
0,001	94,36	97,34	0,0799
0,0001	92,66	94,50	0,1409
0,00001	65,29	77,65	0,7488

Berdasarkan Tabel 3 dapat dilihat penggunaan *learning rate* yang terlalu besar juga terlalu kecil dapat menurunkan nilai akurasi *training* maupun akurasi *validation*, serta terjadinya *underfitting* pada *learning rate* dengan nilai 0.1 dan 0.01. Hal tersebut dikarenakan semakin besar *learning rate*, maka semakin cepat proses *training* model, tetapi tingkat akurasi akan menurun. Hasil menunjukkan bahwa *learning rate* 0.001(10^{-3}) menghasilkan akurasi *validation* tertinggi sebesar 97,34 %.

Percobaan Variasi Jenis Optimizer

Skenario ketiga dilakukan percobaan dengan beberapa jenis *optimizer* seperti RMSprop, SGD, Adam, dan Adamax. Pada saat melakukan percobaan ini terdapat beberapa parameter lain seperti *epoch* yang digunakan 100, *batch size* bernilai 32 dan *learning rate* 0.001. Hasil percobaan disajikan pada Tabel 4.

Tabel 4. Hasil Perbandingan Jenis Optimizer

Optimizer	Akurasi Training	Akurasi Validation	Loss Validation
RMSprop	95,69	96,59	0,1086
SGD	36,03	54,16	1,6881
Adam	94,36	97,34	0,0799
Adamax	90,57	94,12	0,1606

Berdasarkan Tabel 4, *optimizer* adam memperoleh nilai akurasi *validation* terbaik sebesar 97,34% dengan nilai *loss validation* terendah. *Optimizer* RMSprop dan Adamax menghasilkan performa yang mendekati *optimizer* Adam, sedangkan *optimizer* SGD memiliki performa terendah dengan akurasi *validation* sebesar 54,16%.

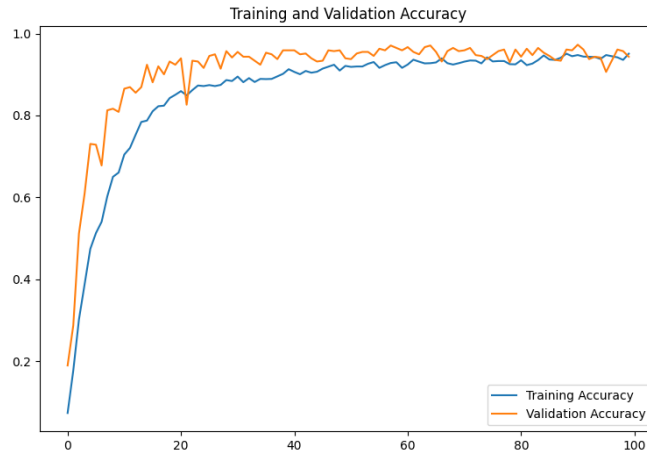
Percobaan Variasi Nilai Batch Size

Skenario keempat melakukan percobaan menggunakan beberapa nilai *batch size* seperti 16, 32, dan 64 untuk mengetahui performa terbaik dari nilai *batch size* yang digunakan. Pada proses percobaan ini terdapat parameter lain seperti *epoch* yang digunakan 100, *optimizer* Adam dengan *learning rate* 0.001. Hasil percobaan dapat dilihat pada Tabel 5.

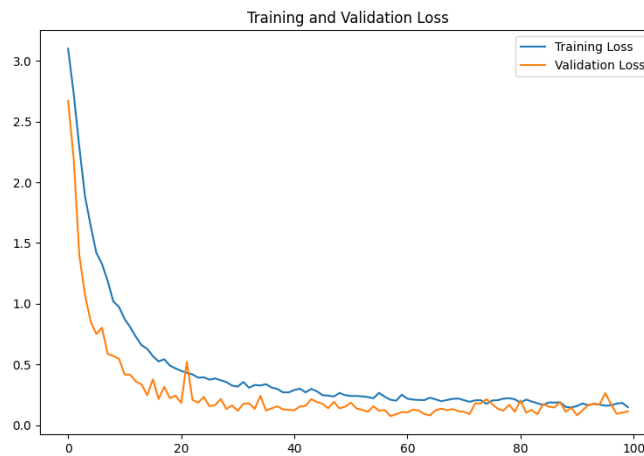
Tabel 5. Hasil Perbandingan Nilai Batch Size

Batch Size	Akurasi Training	Akurasi Validation	Loss Validation
16	93,27	96,78	0,1409
32	94,36	97,34	0,0799
64	95,97	96,59	0,1017

Berdasarkan Tabel 5, dapat dilihat bahwa saat *training* model menggunakan *batch size* 32 menghasilkan akurasi *validation* sebesar 97,34%, selisih sedikit dari *batch size* lainnya yang memperoleh 96%. Pada Gambar 13 dan Gambar 14 merupakan hasil dari percobaan terbaik saat pelatihan model menggunakan arsitektur CNN seperti Gambar 9. Model tidak mengalami *overfitting* dilihat dari grafik tersebut. Akurasi validasi meningkat dan *loss* validasi menurun pada 20 *epoch* pertama, untuk dari *epoch* ke-21 dan seterusnya cenderung stabil.



Gambar 12. Grafik Akurasi



Gambar 13. Grafik Loss

Akurasi validasi mencapai tingkat tertinggi sebesar 97,34%, sementara itu akurasi *training* mencapai 94,36%. Kemudian pada *epoch* yang sama, *loss* validasi mencapai titik terendah sebesar 8%.

5. Penutup

Pada penelitian ini, sebuah model CNN berhasil dikembangkan dengan kemampuan yang baik dalam mengenali tulisan tangan aksara Bima. Hasil optimalisasi menggunakan kombinasi *hyperparameter* dengan nilai *epoch* 100, *learning rate* 0,001, *optimizer* Adam, dan *batch size* 32 mendapatkan nilai akurasi untuk data uji sebesar 97,34%, *precision* 97,56%, *recall* 97,34%, dan *f1-score* 97,31%. Dengan hasil tersebut, diharapkan bahwa model dapat diterapkan dalam *website* maupun aplikasi *mobile* untuk media pembelajaran juga sebagai salah satu upaya pelestarian aksara Bima. Disarankan untuk penelitian selanjutnya memperbanyak data citra asli dan melakukan perbandingan dengan arsitektur CNN lainnya seperti GoogleNet, ResNet18, ResNet50, dan ShuffleNet agar mendapatkan akurasi yang lebih tinggi.

Referensi

- [1] Canberra, "Indonesia." Diakses: 6 Februari 2024. [Daring]. Tersedia pada: <https://www.kemlu.go.id/canberra/id/read/indonesia/2186/etc-menu>
- [2] M. Alfian Tuflih, Mayong, dan Nensilianti, "Pelatihan Membaca dan Menulis Aksara Bima Siswa SMK Negeri 1 Kota Bima," *PENGABDI: Jurnal Hasil Pengabdian Masyarakat*, vol. 3, no. 1, hlm. 1–12, 2022, doi: 10.26858/pengabdi.v3i1.33051.

- [3] Munawar Sulaiman, "Aksara Mbojo." Diakses: 30 Oktober 2023. [Daring]. Tersedia pada: <https://warisanbudaya.kemdikbud.go.id/?newdetail&detailCatat=6561>
- [4] Walikota Bima, *Peraturan Walikota Kota Bima Nomor 50 Tahun 2019 tentang Penetapan Mata Pelajaran Bahasa, Sejarah, Seni Budaya Dan Ketrampilan Sebagai Mata Pelajaran Muatan Lokal Untuk Sekolah Dasar Dan Sekolah Menengah Pertama Di Kota Bima*. Bima, Indonesia: <https://peraturan.bpk.go.id/Details/129985/perwali-kota-bima-no-50-tahun-2019>, 2019.
- [5] F. Bimantoro, A. Aranta, G. Satya Nugraha, R. Dwiyanaputra, dan A. Yudo Husodo, "Pengenalan Pola Tulisan Tangan Aksara Bima menggunakan Ciri Tekstur dan KNN (Handwriting Recognition of Bima Script using Texture Features and KNN)," *Journal of Computer Science and Informatics Engineering*, vol. 5, no. 1, hlm. 60–67, 2021, doi: 10.29303/jcosine.v5i1.387.
- [6] M. I. Fidatama, "Pengenalan Pola Tulisan Tangan Aksara Bima Menggunakan Metode Ekstraksi Ciri Local Binary Pattern, Metode Reduksi Data Latih KSupport Vector Nearest Neighbour, Dan Metode Klasifikasi K-Nearest Neighbour," Thesis, Universitas Mataram, Mataram, 2021.
- [7] R. Aryanto dan M. Alfian Rosid, "Penerapan Deep Learning untuk Pengenalan Tulisan Tangan Bahasa Akasara Lota Ende dengan Menggunakan Metode Convolutional Neural Networks (CNN)," *Jurnal Informasi dan Teknologi*, vol. 5, no. 4, hlm. 258–264, 2023, doi: 10.37034/jidt.v5i1.313.
- [8] G. A. Noor, D. I. Mulyana, dan F. Akbar, "Optimasi Image Classification Pada Burung Kenari Dengan Menggunakan Data Augmentasi dan Convolutional Neural Network," *Smart Comp : Jurnalnya Orang Pintar Komputer*, vol. 11, no. 2, hlm. 226–238, 2022, doi: 10.30591/smartcomp.v11i2.3530.
- [9] D. Darmawan, "Implementasi Metode Convolutional Neural Network (CNN) Dalam Mendeteksi Jenis Sampah," Thesis, Universitas Jambi, Jambi, 2023.
- [10] A. M. T. Andar, N. Fadillah, dan Munawir, "Pengenalan Tulisan Tangan Karakter Aksara Batak Toba dengan Metode Convolutional Neural Network (CNN)," *Jurnal Edukasi dan Penelitian Informatika*, vol. 9, no. 2, hlm. 242–252, 2023, doi: 10.26418/jp.v9i2.64242.
- [11] A. Maharil, "Perbandingan Arsitektur VGG16 Dan ResNet50 Untuk Rekognisi Tulisan Tangan Aksara Lampung," *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 3, no. 2, hlm. 236–243, 2022, doi: 10.33365/jatika.v3i2.2030.
- [12] A. Willyanto, D. Alamsyah, dan H. Irsyad, "Identifikasi Tulisan Tangan Aksara Jepang Hiragana Menggunakan Metode CNN Arsitektur VGG-16," *Jurnal Algoritme*, vol. 2, no. 1, hlm. 1–11, 2021, doi: 10.35957/algoritme.v2i1.1450.
- [13] T. Q. Vinh, L. H. Duy, dan N. T. Nhan, "Vietnamese handwritten character recognition using convolutional neural network," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 2, hlm. 276–283, Jun 2020, doi: 10.11591/ijai.v9.i2.pp276-283.
- [14] I. Khandokar, M. Hasan, F. Ernawan, S. Islam, dan M. N. Kabir, "Handwritten character recognition using convolutional neural network," *J Phys Conf Ser*, vol. 1918, no. 4, hlm. 1–5, Jun 2021, doi: 10.1088/1742-6596/1918/4/042152.
- [15] M. Junihardi, S. Sanjaya, L. Handayani, dan F. Syafria, "Klasifikasi Daging Sapi Dan Daging Babi Menggunakan Arsitektur EfficientNet-B3 Dan Augmentasi Data," *Jurnal TEKINKOM*, vol. 6, no. 1, hlm. 16–25, 2023, doi: 10.37600/tekinkom.v6i1.845.
- [16] M. D. Nadarajan, S. Raghava, S. Giri, dan B. Kumar Depuru, "Enhancing Warehouse Operations Through Artificial Intelligence: Pallet Damage Classification with Deep Learning Insights," *Int J Innov Sci Res Technol*, vol. 8, no. 11, hlm. 2556–2563, 2023, doi: 10.5281/zenodo.10391024.
- [17] R. Adam, "Image Classification Dengan Cnn Dan Tensorflow." Diakses: 8 Februari 2024. [Daring]. Tersedia pada: <https://structilmy.com/blog/2021/01/18/image-classification-dengan-cnn-dan-tensorflow/>

- [18] A. A. Asiri *dkk.*, “Block-Wise Neural Network for Brain Tumor Identification in Magnetic Resonance Images,” *Computers, Materials and Continua*, vol. 73, no. 3, hlm. 5735–5753, 2022, doi: 10.32604/cmc.2022.031747.
- [19] O. A. Shawky, A. Hagag, E. S. A. El-Dahshan, dan M. A. Ismail, “Remote sensing image scene classification using CNN-MLP with data augmentation,” *Optik (Stuttg)*, vol. 221, Nov 2020, doi: 10.1016/j.ijleo.2020.165356.
- [20] A. M. Saleh dan T. Hamoud, “Analysis And Best Parameters Selection For Person Recognition Based On Gait Model Using CNN Algorithm And Image Augmentation,” *J Big Data*, vol. 8, no. 1, Des 2021, doi: 10.1186/s40537-020-00387-6.
- [21] Y. Tian, “Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm,” *IEEE Access*, vol. 8, hlm. 125731–125744, 2020, doi: 10.1109/ACCESS.2020.3006097.
- [22] G. Y. Christiawan, R. A. Putra, A. Sulaiman, E. Poerbaningtyas, dan S. W. Putri Listio, “Penerapan Metode Convolutional Neural Network (CNN) Dalam Mengklasifikasikan Penyakit Daun Tanaman Padi,” *J-INTECH*, vol. 11, no. 2, hlm. 294–306, Des 2023, doi: 10.32664/j-intech.v11i2.1006.
- [23] A. Peryanto, A. Yudhana, dan D. R. Umar, “Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network,” *Format : Jurnal Ilmiah Teknik Informatika*, vol. 8, no. 2, hlm. 138–147, 2019, doi: 10.22441/format.2019.v8.i2.007.
- [24] S. Dahiya, T. Gulati, dan D. Gupta, “Performance Analysis Of Deep Learning Architectures For Plant Leaves Disease Detection,” *Measurement: Sensors*, vol. 24, Des 2022, doi: 10.1016/j.measen.2022.100581.
- [25] D. Cindy Agustin, M. Alfian Rosid, dan N. Ariyanti, “Implementasi Convolutional Neural Network Untuk Deteksi Kesegaran Pada Apel,” *Jurnal FASILKOM (teknologi inFormASi dan Ilmu KOMputer)*, vol. 13, no. 2, hlm. 145–150, 2023, doi: 10.37859/jf.v13i02.5175.
- [26] A. S. Paymode dan V. B. Malode, “Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG,” *Artificial Intelligence in Agriculture*, vol. 6, hlm. 23–33, Jan 2022, doi: 10.1016/j.aiaa.2021.12.002.
- [27] M. Rafly Alwanda, R. Putra, K. Ramadhan, dan D. Alamsyah, “Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle,” *Jurnal Algoritme*, vol. 1, no. 1, hlm. 45–56, 2020, doi: 10.35957/algoritme.v1i1.434.
- [28] K. O. Afebu, Y. Liu, dan E. Papatheou, “Feature-based intelligent models for optimisation of percussive drilling,” *Neural Networks*, vol. 148, hlm. 266–284, Apr 2022, doi: 10.1016/j.neunet.2022.01.021.
- [29] M. Yunus, M. Husni, dan M. M. Mufadhdhal, “Klasifikasi Sentimen Terhadap Badan Penyelenggara Jaminan Sosial (BPJS) Pada Media Sosial Twitter Menggunakan Naive Bayes,” *SMATIKA JURNAL*, vol. 11, no. 02, hlm. 81–91, Des 2021, doi: 10.32664/smatika.v11i02.577.
- [30] E. A. Pranata, S. Subari, dan G. F. Gunawan, “Penerapan Metode Naive Bayes Untuk Klasifikasi Sms Spam Menggunakan Java Rogramming,” *J-INTECH*, vol. 7, no. 02, hlm. 104–108, Des 2019, doi: 10.32664/j-intech.v7i02.435.
- [31] K. Nugroho dan F. N. Hasan, “Analisis Sentimen Masyarakat Mengenai RUU Perampasan Aset Di Twitter Menggunakan Metode Naive Bayes,” *SMATIKA JURNAL*, vol. 13, no. 02, hlm. 273–283, Des 2023, doi: 10.32664/smatika.v13i02.899.
- [32] M. Yildirim dan A. Cinar, “Classification of Alzheimer’s disease MRI images with CNN based hybrid method,” *Ingenierie des Systemes d’Information*, vol. 25, no. 4, hlm. 413–418, Agu 2020, doi: 10.18280/isi.250402.
- [33] K. Muhammad, T. Hussain, dan S. W. Baik, “Efficient CNN based summarization of surveillance videos for resource-constrained devices,” *Pattern Recognit Lett*, vol. 130, hlm. 370–375, Feb 2020, doi: 10.1016/j.patrec.2018.08.003.

- [34] KantinIT, "Confusion Matrix: Pengertian, Cara Kerja dan Contoh Soal." Diakses: 17 Februari 2024. [Daring]. Tersedia pada: <https://kantinit.com/kecerdasan-buatan/confusion-matrix-pengertian-cara-kerja-dan-contoh-soal/>