Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 23, No. 2, March 2024, pp. 453∼464
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v23i2.xxx

DenseNet Architecture for Efficient and Accurate Recognition of Javanese Script Hanacaraka Character

Egi Dio Bagus Sudewo, Muhammad Kunta Biddini, Abdul Fadlil
Universitas Ahmad Dahlan,Yogyakarta, Indonesia

Article Info

Article history:

Received February 07, 2024
Revised March 18, 2024
Accepted March 28, 2024

Keywords:

Convolution Neural Network
Character Recognition
Cultural Heritage Preservation
Densenet
Javanese Hanacaraka Script

ABSTRACT

This study introduced a specifically optimized DenseNet architecture for recognizing Javanese Hanacaraka characters, focusing on enhancing efficiency and accuracy. The research aimed to preserve and celebrate Java's rich cultural heritage and historical significance through the development of precise character recognition technology. The method used advanced techniques within convolutional neural networks (CNN) to integrate feature extraction across densely connected layers efficiently. The result of this study was that the developed model achieved a training accuracy of 100% and a validation accuracy of approximately 99.50% after 30 training epochs. Furthermore, when tested on previously unseen datasets, the model exhibited exceptional accuracy, precision, recall, and F1-score, reaching 100%. These findings underscored the remarkable capability of DenseNet architecture in character recognition, even across novel datasets, suggesting significant potential for automating Javanese Hanacaraka text processing across various applications, ranging from text recognition to digital archiving. The conclusion drawn from this study suggests that optimizing DenseNet architecture can be a significant step in preserving and developing character recognition technology for Javanese Hanacaraka while supporting broader applications within the context of Javanese culture and history.

Corresponding Author:

Egi Dio Bagus Sudewo, +6288264240687,
Master's Degree in Informatics,
Universitas Ahmad Dahlan, Yogyakarta, Indonesia.
Email: egidio8gali@gmail.com

How to Cite:
E. D. Sudewo, M. Biddinika, and A. Fadlil, "DenseNet Architecture for Efficient and Accurate Recognition of Javanese Script Hanacaraka Character", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 23, No. 2, pp. 453-464, Mar, 2024.

1. INTRODUCTION

The emergence of digital image processing technology has opened new avenues in various fields, particularly in cultural heritage preservation, through efficient and accurate character recognition. One primary focus in this domain is recognizing characters in the Javanese script Hanacaraka, a traditional writing system deeply ingrained in Javanese cultural wealth and steeped in profound historical significance. Hanacaraka represents a unique and valuable form of writing within the Javanese cultural context, serving as a means of communication and a symbol of rich cultural identity and extensive history. Each character in the Javanese script, Hanacaraka, possesses distinct shapes and carries profound meanings, often reflecting Javanese cultural values and philosophies [1]. These characters are integral to Javanese culture and history, serving as more than just communication tools but also as symbols of cultural identity. Therefore, character recognition in the Javanese script Hanacaraka is important in preserving and understanding Javanese cultural heritage [2].

In this contemporary digital era, the importance of character recognition in Javanese script Hanacaraka has increased, especially in cultural heritage preservation and dissemination. Despite significant advancements in character recognition, some challenges remain, particularly in the Javanese script Hanacaraka recognition domain, characterized by its complex structure and unique features, as in Figure 1. One of the industry-standard approaches to character recognition is Convolutional Neural Networks (CNNs) [3]. CNNs are artificial neural networks inspired by how humans process visual information [4]. They utilize convolutional processes to extract features from images and learn complex patterns such as character recognition [5], alphanumeric pattern recognition [6], handwriting recognition [7], and recognition of other complex patterns in raw data form [8]. Thus, CNNs have become the preferred choice in various character recognition applications due to their ability to learn complex patterns from images [9].

Some of the differences between this research and previous research studies utilizing CNNs include research [10] that compared pooling methods for Javanese script characters and obtained results of 93% for average pooling and 92% for max pooling using a 5x5 filter. Moreover, A. Budiman et al. identified Javanese script patterns using the Ngaglena script and Histogram Chain code method and achieved an accuracy of 76%. However, the current research focuses on the Javanese script Hanacaraka, known as the classical Javanese script, which presents a unique challenge [11]. Research [12] investigated Javanese script characters using the random forest algorithm with 6000 data points and obtained accuracy, precision, and recall results of 97.7%, while the current study employs 12000 data points.

Characters in Javanese script indeed present fascinating research subjects. However, in the context of character recognition in the Javanese script Hanacaraka, the utilization of DenseNet architecture becomes an intriguing point of interest. DenseNet is a type of artificial neural network architecture characterized by dense connections between its layers. Unlike CNNs, which utilize sparse connections between layers, DenseNet connects each layer directly to one another, enabling more efficient information flow and reinforcing feature learning. With this unique structure, DenseNet offers the potential to enhance character recognition performance, especially in cases involving Javanese Hanacaraka script with its structural complexity and uniqueness. Therefore, this study aims to explore the use of DenseNet in this context to provide a deeper understanding and better results in Javanese Hanacaraka script character recognition. The contribution of this research to the development of science lies in the increased understanding of DenseNet's potential in character recognition, particularly in the Javanese Hanacaraka script. Additionally, this research may offer practical benefits in developing character recognition technology for specific languages and cultures while encouraging further research in the field of character recognition based on distinctive scripts.

2. RESEARCH METHOD

This research employs a quantitative methodology, utilizing image data as the subject for computation using the CNN method with DenseNet architecture and a learning rate of 0.001, with Adam optimizer as an additional component. The study utilizes Google Colab and various libraries as supporting tools.

2.1. Data Source

The study employs the Convolutional Neural Network (CNN) method with DenseNet121 architecture for recognizing Javanese script Hanacaraka characters. The dataset is available via Kaggle Hanna Hunafa and comprises 20 labels of Javanese script Hanacaraka characters, totaling 8,400 training data, 2,400 test data, and 1,200 validation data. All data are in the form of PNG images in grayscale format.

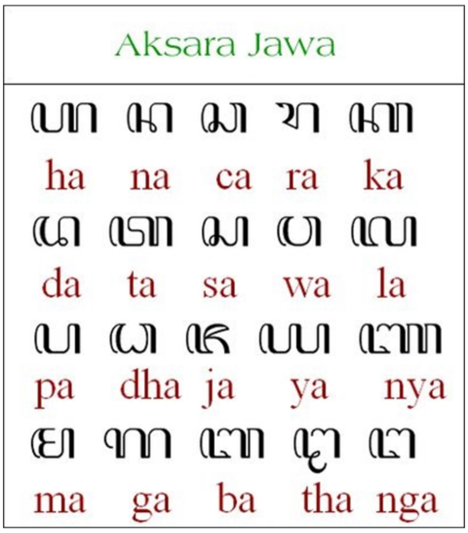

Figure 1. Javanese Script Hanacaraka (Image Adopted From pinterest.com)

2.2. Research Steps

The research begins with a literature review search as the first step to comprehend the theoretical foundations and related studies. The subsequent step involves collecting a suitable dataset aligned with the research objectives. This dataset then undergoes preprocessing to prepare it according to the analysis requirements. The Convolutional Neural Network (CNN) algorithm is implemented, utilizing the DenseNet architecture to extract features from the preprocessed dataset. The next process entails training the CNN model using the preprocessed dataset. Once the model is trained, testing is conducted to evaluate the model's performance. The prediction results from the model are evaluated to determine accuracy and overall performance. The output of this research is the prediction results representing the model's effectiveness in solving the investigated problem, as elaborated in Figure 2.

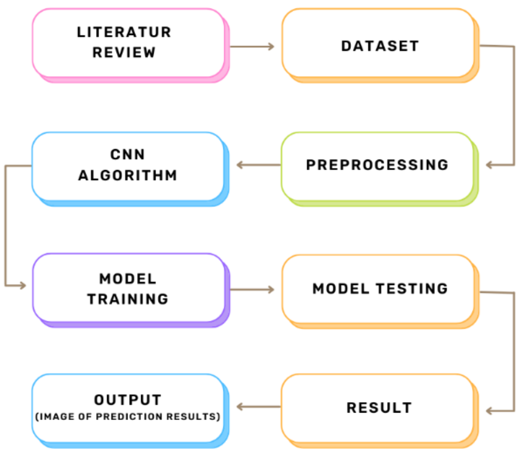

Figure 2. Flowchart of Research Stages

2.3. Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of artificial neural network specifically designed for image recognition problems inspired by biological visual recognition processes [13]. CNN consists of multiple layers, including an input layer, convolutional layer, pooling layer, fully connected layer, and output layer [14], as shown in Figure 3. The convolutional layer, as the core of CNN, plays a crucial role in extracting image features. By scanning the image using convolutional kernels, the convolutional layer can utilize information from neighboring regions in the image to extract relevant features [15]. Commonly used CNN architectures include ResNet-18 [16], MobileNet [17], VGG16 [18], AlexNet [19], and InceptionV3 [20], among others [21].

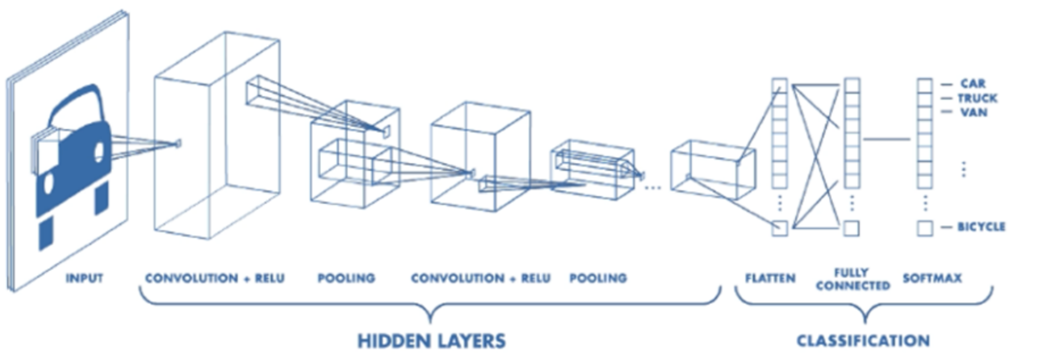

Figure 3. Architecture of CNN

2.4. DenseNet

The DenseNet architecture is a well-known artificial neural network architecture notable for its dense connections between layers [22]. This means that each layer in the DenseNet architecture receives input not only from the previous layer but also from all previous layers in the network, as shown in Figure 4. In other words, each layer directly connects to every preceding layer. This creates a highly dense and interconnected structure within the network. The DenseNet architecture facilitates a more efficient and robust information flow across the network. Because each layer has direct access to information from previous layers, the network can leverage features extracted from all levels of image resolution, from early layers to deeper layers. This helps prevent information loss during forward propagation and can enhance the network's overall performance [23].

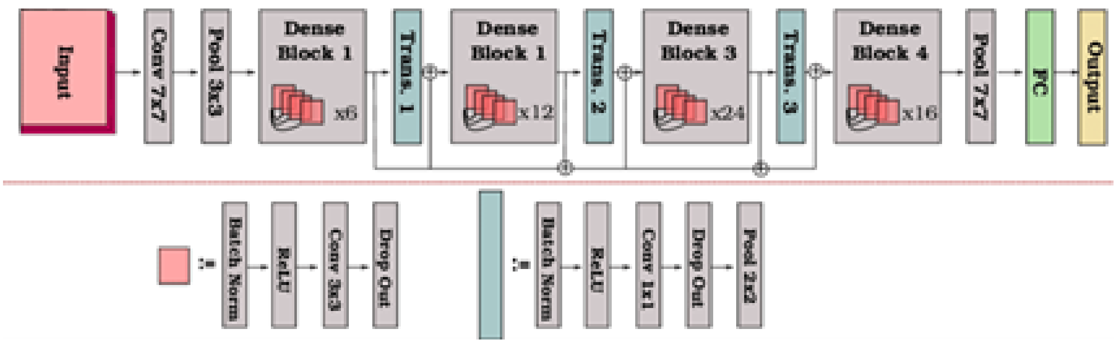

Figure 4. Architecture of DenseNet

Additionally, DenseNet's compact architecture significantly reduces the number of parameters required for the network to function, as some features are shared across various layers. This can alleviate overfitting issues and enable training with smaller datasets [24]. In this study, the researcher utilized DenseNet121, a basic variant of the DenseNet architecture consisting of 121 layers, as depicted in Figure 5 [25]. In practice, DenseNet is commonly employed in tasks related to image recognition, such as image classification, object detection, and image segmentation. Its main advantage in these tasks lies in its ability to learn highly robust feature representations from input images using efficient dense connections [26].

```
==========================================================================================
Layer (type:depth-idx)                        Output Shape              Param #
==========================================================================================
DenseNetModel                                 [1, 20]                   --
├─DenseNet: 1-1                               [1, 20]                   --
│    └─Sequential: 2-1                        [1, 1024, 2, 2]           --
│    │    └─Conv2d: 3-1                       [1, 64, 32, 32]           9,408
│    │    └─BatchNorm2d: 3-2                  [1, 64, 32, 32]           128
│    │    └─ReLU: 3-3                         [1, 64, 32, 32]           --
│    │    └─MaxPool2d: 3-4                    [1, 64, 16, 16]           --
│    │    └─_DenseBlock: 3-5                  [1, 256, 16, 16]          335,040
│    │    └─_Transition: 3-6                  [1, 128, 8, 8]            33,280
│    │    └─_DenseBlock: 3-7                  [1, 512, 8, 8]            919,680
│    │    └─_Transition: 3-8                  [1, 256, 4, 4]            132,096
│    │    └─_DenseBlock: 3-9                  [1, 1024, 4, 4]           2,837,760
│    │    └─_Transition: 3-10                 [1, 512, 2, 2]            526,336
│    │    └─_DenseBlock: 3-11                 [1, 1024, 2, 2]           2,158,080
│    │    └─BatchNorm2d: 3-12                 [1, 1024, 2, 2]           2,048
│    └─Linear: 2-2                            [1, 20]                   20,500
==========================================================================================
Total params: 6,974,356
Trainable params: 6,974,356
Non-trainable params: 0
Total mult-adds (M): 231.38
==========================================================================================
Input size (MB): 0.05
Forward/backward pass size (MB): 14.74
Params size (MB): 27.90
Estimated Total Size (MB): 42.68
==========================================================================================
```

Figure 5. Densenet Parameters of This Research

2.5. Preprocessing

In this research, the processing process changes the image size to 64x64 pixels. This is done to ease training and simplify the model training process. Preprocessing of each data improves application performance [27].

2.6. Evaluation

Evaluation is vital in assessing the performance of data mining techniques. Commonly, classification models are evaluated using a confusion matrix to calculate accuracy. This matrix helps measure test results based on the dataset, aiding in computing accuracy, defined as the proportion of correct classification outcomes among the total evaluated dataset, including true positives and true negatives [26].

a. Accuracy: Accuracy emerges as the primary metric for evaluating the performance of our deep-learning classifiers. It is computed in Equation (1) by adding true positives and true negatives and then dividing by the total sum of the confusion matrix components. While having a dependable model is paramount, it is equally critical to maintain balanced datasets with nearly equivalent false positive and false negative values. Hence, the components of the confusion matrix mentioned earlier are vital for evaluating the accuracy of Javanese script recognition.

$$Accuracy\,(\%) = \frac{TP + TP}{TP + FP + FN + TN} * 100 \tag{1}$$

b. Precision: Equation (2) describes the correlation between true positive predictions and the overall positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

c. Recall: In Equation (3), Recall, also referred to as sensitivity, indicates the proportion of true positive predictions relative to the combined total of predicted true positives and false negatives.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

d. F1-score: The F1-score, represented as (4), acts as a holistic gauge of the model's accuracy by combining precision and recall. It is twice the ratio of the product to the sum of precision and recall metrics.

$$F1 - score = 2\left(\frac{Precision\ x\ Recall}{Precision + Recall}\right) \tag{4}$$

3. RESULT AND ANALYSIS

This study utilizes Google Colab to build the model, leveraging various supporting libraries such as Torch and Sklearn. Additionally, Torchvision is employed for preprocessing tasks such as cropping and resizing, among others. Initially, the data undergoes Torchvision preprocessing before being fed into the model. The model training process is carried out through an iterative process known as epochs, where each epoch involves processing the entire training dataset once. During the training process using CNN with DenseNet architecture, various metrics are evaluated for each epoch, including the time taken for each epoch, training accuracy, training loss, validation accuracy, and validation loss. Figure 6 in the research report illustrates the final results from epoch 30 of the training process. This research yielded highly impressive results at epoch 30, utilizing 8400 training data, 2400 testing data, and 1200 validation data as the dataset, with the dataset size transformed into 64x64 pixels, using a learning rate of 0.001 and Adam optimizer.

```
Epoch 30/30 - Training: 100%|██████████| 132/132 [00:36<00:00,  3.66it/s]
Epoch 30/30, Training Loss: 0.0000, Training Accuracy: 1.0000
Validation: 100%|██████████| 38/38 [00:09<00:00,  4.07it/s]
Validation Loss: 0.0106, Validation Accuracy: 0.9950
Testing: 100%|██████████| 19/19 [02:38<00:00,  8.32s/it]
Test Accuracy: 1.0000
Test Precision: 1.0000
Test Recall: 1.0000
Test F1-score: 1.0000
```

Figure 6. Result from Research

The model's performance during training, validation, and testing was exceptional. In the 30th epoch, the model achieved 100% accuracy in training. This can be observed in the confusion matrix diagram. Figure 7 indicates its ability to "memorize" the Javanese script Hanacaraka training data effectively.

Classes	ba	ca	da	dha	ga	ha	ja	ka	la	ma	na	nga	nya	pa	ra	sa	ta	tha	wa	ya
ba	60																			
ca		60																		
da			60																	
dha				60																
ga					70															
ha						60														
ja							60													
ka								60												
la									60											
ma										60										
na											60									
nga												60								
nya													60							
pa														60						
ra															60					
sa																60				
ta																	60			
tha																		60		
wa																			60	
ya																				60

Figure 7. Matrix Confusion

During the validation process, the model demonstrated exceptional performance with an accuracy of 99.50%, showcasing its proficiency in classifying new data while minimizing significant overfitting risks. This achievement can be attributed to preprocessing steps, particularly the resizing of the dataset, as depicted in Figure 8, which likely facilitated better model generalization. Furthermore, when tested with previously unseen test data, the model exhibited flawless results, achieving perfect accuracy, precision, and an F1 score of 100%. These remarkable outcomes were further validated through meticulous matrix calculations presented in Figure 7, utilizing Equations (1-4), and visually depicted in the training line graph in Figure 8. Such findings confirm the model's effectiveness and highlight its robustness in handling various data types. Additionally, this study emphasizes the superiority of the CNN approach with DenseNet architecture over traditional methods, as evidenced by surpassing previous research benchmarks in Javanese character recognition. This accomplishment necessitates further analysis to identify potential patterns in recognition accuracy and potential sources of errors that could impact the model's performance. Such insights may include identifying character classes with lower recognition accuracy or patterns in recognition requiring further investigation. Additionally, addressing challenges encountered during testing is crucial to understanding when the model may not deliver optimal results, such as difficulties in recognizing characters or patterns in test data affecting performance. There might also be a risk of overfitting due to limited data if the training epochs are too high. Through comprehensive analysis, we can gain deeper insights into the strengths and weaknesses of the DenseNet model in Javanese character recognition, guiding future development and improvement.

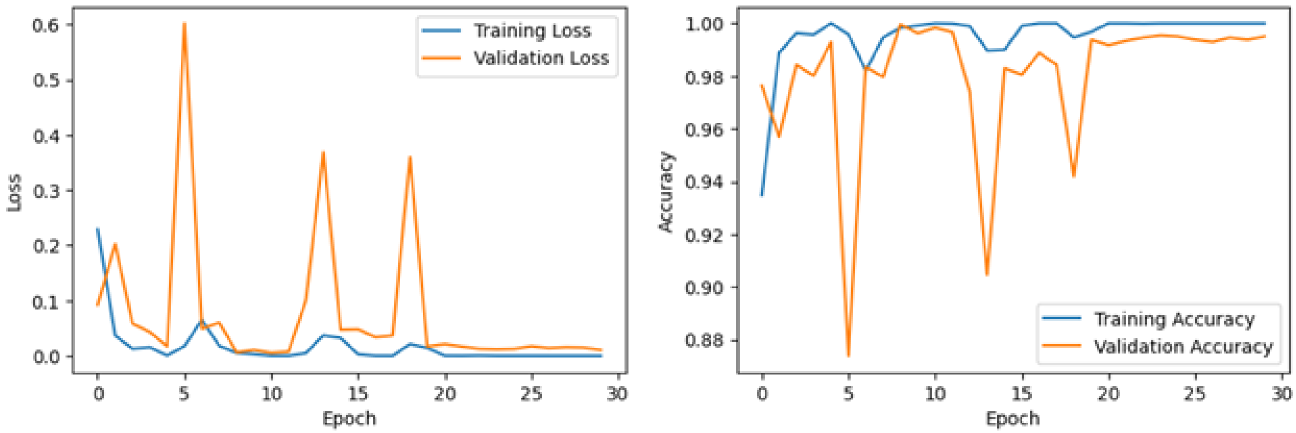

Figure 8. Line Graph of Training and Validation Results

Furthermore, this underscores that the CNN method with DenseNet architecture surpasses comparisons with methods like CNN by just pooling or using other methods in previous research for recognizing Javanese characters [9–11], as described in Table 1. Based on the comparison in Table 1, the findings in this research are that the CNN method with DenseNet architecture can recognize Javanese script letter objects. This is in line with previous research, which used the same method. This research shows better results than previous research.

Table 1. Comparison of Previous Research with This Research

Research Title	Researcher Name, Year	Object of Research	Method	Result
Identification of Learning Javanese Script Handwriting Using Histogram Chain Code	Budiman, Arif, Fadlil, Abdul, Umar, Rusydi 2023 [11]	Javanese Ngaglena Script	Histogram Chain Code	The accuracies of these models vary, with the highest accuracy achieved by the HCC model with 72 parameters and images divided into 9 parts, reaching 83%.
Classification of handwritten Javanese script using random forest algorithm	Rasyidi, Mohammad Arif Bariyah, Taufiqotul Riskajaya, Yohanes Indra Septyani, Ayunda Dwita, 2021 [12]	Javanese Hanacaraka Script	Random Forest Algorithm	From the experimental results, it is evident that the generated random forest model is capable of classifying Javanese script characters very accurately, with accuracy, precision, and recall reaching 97.7%.

Continued on the next page

Continuation from the previous page

Research Title	Researcher Name, Year	Object of Research	Method	Result
Pooling Comparison in CNN Architecture for Javanese Script Classification	Muhdalifah, Mujastia Feliati, 2021 [10]	Javanese Hanacaraka Script	Pooling Comparison in CNN	Based on the experimental results, average pooling can yield higher accuracy compared to max-pooling layers. Furthermore, average pooling also achieves faster training times. It can be concluded that average pooling produces the best Javanese script classification performance.
DenseNet Architecture for Efficient and Accurate Recognition of Javanese Script Hanacaraka Character	Egidio, Kunta, Fadlil, 2024	Javanese Hanacaraka Script	CNN with DenseNet Architecture	After 30 training epochs, our model achieved a remarkable 100% training accuracy and around 99.50% validation accuracy with careful training methods and customized data augmentation procedures. Surprisingly, our model demonstrated exceptional overall accuracy, precision, recall, and F1-score, all reaching 100%, when evaluated on unpublished datasets.

Table 2 presents several examples of prediction results for Javanese script letters. Out of the 20 letters, the researcher showcases predictions for 5 letters: Ba, Ca, Da, Dha, and Ga. This table offers insight into the model's performance by showcasing the predicted outcomes for a selection of input samples. These predictions are crucial for demonstrating the effectiveness and accuracy of the model in recognizing and classifying Javanese characters.

Table 2. Sample Prediction Result

Character	Sample Character images	Prediction Results
Ba		
Ca		
Da		
Dha		

Continued on the next page

Continuation from the previous page

Character	Sample Character images	Prediction Results
Ga		

4. CONCLUSION

This study demonstrates that the utilization of DenseNet architecture for recognizing Javanese Hanacaraka characters has yielded highly impressive results, with training and validation accuracies reaching 100% and 99.50%, respectively. The testing results also indicate excellent performance, with accuracy, precision, recall, and F1 score all achieving 100%. Implications of this research encompass a better understanding of character recognition technology and the need for further exploration of integrating this technology into everyday societal use. Suggestions for further development include exploring larger datasets, practical applications such as handwriting recognition, model performance optimization, and further research using Javanese Hanacaraka characters in modern applications. Consequently, this study has the potential to significantly contribute to the development of efficient and accurate Javanese Hanacaraka character recognition technology. However, it is important to acknowledge research limitations such as constraints in dataset availability, model performance, or result generalizability, underscoring the necessity for additional steps to address these limitations in future research.

5. ACKNOWLEDGEMENTS

Completing this research would not have been possible without the assistance of various individuals. The author expresses gratitude to the Chancellor of Ahmad Dahlan University, as well as to Supervising Professor 1 and Supervising Professor 2, whose support was invaluable in conducting this research.

6. DECLARATIONS

AUTHOR CONTIBUTION

The first author's contribution is to design the overall research outline and seek more knowledge about the CNN method with DenseNet architecture. The second author provides input regarding the journal's contents, and the third author provides input regarding journal writing.

FUNDING STATEMENT

-

COMPETING INTEREST

The author declares that there are no conflicts or interests between the editors and reviewers regarding the publication of this article.

REFERENCES

[1] F. K. Sari, "The Local Wisdom in Javanese Thinking Culture," *Diksi*, vol. 28, no. 1, pp. 86–100, 2020.

[2] J. Sukoyo, E. S. Utami, and E. Kurniati, "The Development of Montessori-Based Javanese Script Learning Model," *Proceedings of the 2nd International Conference on Innovation in Education and Pedagogy (ICIEP 2020)*, vol. 619, no. Iciep 2020, pp. 99–102, 2022.

[3] J. D. Kelleher, *The Deep Learning*, 2019.

[4] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Progress in Artificial Intelligence*, vol. 9, no. 2, pp. 85–112, 2020, doi: 10.1007/s13748-019-00203-0.

[5] N. P. Sutramiani, N. Suciati, and D. Siahaan, "Transfer Learning on Balinese Character Recognition of Lontar Manuscript Using MobileNet," *ICICoS 2020 - Proceeding: 4th International Conference on Informatics and Computational Sciences*, pp. 0–4, 2020, doi: 10.1109/ICICoS51170.2020.9299030.

[6] S. K. C and C. Nattee, "Handwritten Alphanumeric Character Recognition," vol. 12, no. 1, pp. 20–30, 2023, doi: 10.5072/FK26H4PV9J.2023.05.08.001.

[7] V. Madane, K. Ovhal, and M. Bhong, "Handwriting Recognition Using Artificial Intelligence Neural Network and Image Processing," *International Research Journal of Modernization in Engineering Technology and Science*, no. 03, pp. 1432–1447, 2023, doi: 10.56726/irjmets34395.

[8] S. F. Müller-Cleve, V. Fra, L. Khacef, A. Pequeño-Zurro, D. Klepatsch, E. Forno, D. G. Ivanovich, S. Rastogi, G. Urgese, F. Zenke, and C. Bartolozzi, "Braille letter reading: A benchmark for spatio-temporal pattern recognition on neuromorphic hardware," *Frontiers in Neuroscience*, vol. 16, 2022, doi: 10.3389/fnins.2022.951164.

[9] U. BS and U. K, "a Review Paper on Ocr Using Convolutional Neural Networks," *International Journal of Engineering Applied Sciences and Technology*, vol. 7, no. 7, pp. 102–106, 2022, doi: 10.33564/ijeast.2022.v07i07.018.

[10] M. F. Muhdalifah, "Pooling Comparison in CNN Architecture for Javanese Script Classification," *International Journal of Informatics and Computation*, vol. 3, no. 2, p. 15, 2022, doi: 10.35842/ijicom.v3i2.30.

[11] A. Budiman, A. Fadlil, and R. Umar, "Identification of Learning Javanese Script Handwriting Using Histogram Chain Code," *EDUMASPUL: Jurnal Pendidikan*, vol. 7, no. 1, pp. 147–153, 2023.

[12] M. A. Rasyidi, T. Bariyah, Y. I. Riskajaya, and A. D. Septyani, "Classification of handwritten javanese script using random forest algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 3, pp. 1308–1315, 2021, doi: 10.11591/eei.v10i3.3036.

[13] D. R. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: a survey," *Evolutionary Intelligence*, vol. 15, no. 1, pp. 1–22, 2022, doi: 10.1007/s12065-020-00540-3.

[14] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, *A survey of the recent architectures of deep convolutional neural networks*. Springer Netherlands, 2020, vol. 53, no. 8.

[15] Purwono, A. Ma'arif, W. Rahmaniar, H. I. K. Fathurrahman, A. Z. K. Frisky, and Q. M. U. Haq, "Understanding of Convolutional Neural Network (CNN): A Review," *International Journal of Robotics and Control Systems*, vol. 2, no. 4, pp. 739–748, 2022, doi: 10.31763/ijrcs.v2i4.888.

[16] S. Singh and D. Schicker, "Seven Basic Expression Recognition Using ResNet-18," pp. 1–3, 2021.

[17] D. Sinha and M. El-Sharkawy, "Thin MobileNet: An Enhanced MobileNet Architecture," *2019 IEEE 10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2019*, pp. 0280–0285, 2019, doi: 10.1109/UEMCON47517.2019.8993089.

[18] D. Gupta, A. Gupta, R. Gandhi, P. Vishwavidyalaya, and S. S. Gupta, "Identification of Alzheimer's disease from MRI image employing a probabilistic deep learning-based approach and the VGG16," 2023.

[19] W. Gong, X. Zhang, B. Deng, and X. Xu, "Palmprint recognition based on convolutional neural network-alexnet," *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019*, vol. 18, pp. 313–316, 2019, doi: 10.15439/2019F248.

[20] N. S. Shadin, S. Sanjana, and N. J. Lisa, "COVID-19 Diagnosis from Chest X-ray Images Using Convolutional Neural Network(CNN) and InceptionV3," *2021 International Conference on Information Technology, ICIT 2021 - Proceedings*, vol. 3, no. September 2012, pp. 799–804, 2021, doi: 10.1109/ICIT52682.2021.9491752.

[21] G. S. Nugraha, M. I. Darmawan, and R. Dwiyansaputra, "Comparison of CNN's Architecture GoogleNet, AlexNet, VGG-16, Lenet -5, Resnet-50 in Arabic Handwriting Pattern Recognition," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 4, no. 2, 2023, doi: 10.22219/kinetik.v8i2.1667.

[22] Z. Zhong, M. Zheng, H. Mai, J. Zhao, and X. Liu, "Cancer image classification based on DenseNet model," *Journal of Physics: Conference Series*, vol. 1651, no. 1, 2020, doi: 10.1088/1742-6596/1651/1/012143.

[23] G. Huang, Z. Liu, G. Pleiss, L. Van Der Maaten, and K. Q. Weinberger, "Convolutional Networks with Dense Connectivity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 8704–8716, 2022, doi: 10.1109/TPAMI.2019.2918284.

[24] Z. Shi, M. Chen, and Z. Wu, "Hyperspectral Image Classification Based on Dual-Scale Dense Network with Efficient Channel Attentional Feature Fusion," *Electronics (Switzerland)*, vol. 12, no. 13, 2023, doi: 10.3390/electronics12132991.

[25] S. A. Albelwi, "Deep Architecture based on DenseNet-121 Model for Weather Image Recognition," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 10, pp. 559–565, 2022, doi: 10.14569/IJACSA.2022.0131065.

[26] E. E.-D. Hemdan, M. A. Shouman, and M. E. Karar, "COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images," 2020.

[27] M. M. Hasan, H. Ali, M. F. Hossain, and S. Abujar, "Preprocessing of Continuous Bengali Speech for Feature Extraction," *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020*, pp. 1–4, 2020, doi: 10.1109/ICCCNT49239.2020.9225469.

[This page intentionally left blank.]