Bulletin of Electrical Engineering and Informatics
Vol. 12, No. 2, April 2023, pp. 860~867
ISSN: 2302-9285, DOI: 10.11591/eei.v12i2.4172

Image preprocessing analysis in handwritten Javanese character recognition

Fetty Tri Anggraeny, Yisti Vita Via, Retno Mumpuni
Department of Informatics, Faculty of Computer Science, Universitas Pembangunan Nasional Veteran Jawa Timur, Surabaya, Indonesia

Article Info

Article history:

Received Jun 2, 2022
Revised Jul 14, 2022
Accepted Sep 30, 2022

Keywords:

Image preprocessing
Image recognition
Javanese character
Morphological process
Neural network

ABSTRACT

The handwriting produced by each person is unique, so each person has a different stroke, even though they write the same letter. Handwritten Javanese is an exciting topic to study, in addition to scientific purposes and preserving Indonesian culture. The Javanese character image dataset is *aksara Jawa*: aksara Jawa custom dataset from the Kaggle database consists of 2,154 train data and 480 evaluation data. This research proposed to analyze the impact of some preprocessing methods in recognizing handwritten Javanese characters. The preprocessing methods are dilation, skeletonization, and noise reduction. The first process is segmentation for region of interest (ROI) extraction, then various preprocessing is used, and finally, the recognition step neural network (NN) to measure the effectiveness of the preprocessing method. The experiment shows that all preprocessing methods (dilation, skeletonization, and noise reduction) give excellent results, especially on the black background color, reaching 98% accuracy. Other experimental findings show that in any preprocessing combination, the black background accuracy is better than the white one.

Corresponding Author:

Fetty Tri Anggraeny
Department of Informatics, Faculty of Computer Science
Universitas Pembangunan Nasional Veteran Jawa Timur
Jl. Raya Rungkut Madya, Gunung Anyar, Surabaya, Indonesia
Email: wichian.sit@mfu.ac.th

1. INTRODUCTION

Some nations, such as Korea, Japan, Thailand, and Indonesia, have the traditional script. Indonesia has more than one traditional script because Indonesia consists of various regional tribes, each tribe has cultural diversity, and some even have tribe letters. For example, the Javanese tribe has a Javanese script, the *Sunda* tribe has *aksara Sunda* [1], Lampung has *aksara Kaganga* [2], Lombok island has *aksara Sasak* [3], Makasar has *aksara Lontara* [4], and Batak tribe has *aksara Batak* [5]. Javanese script consists of 20 letters that come from the legend of *Ajisaka*. In the legend, there is a fight between two servants of *Ajisaka*, namely *Dora* and *Sembada*. Both of them died because they were equally strong. To commemorate his two servants, *Ajisaka* made up the story of his two servants in a series of letters known as Javanese script or *Hanacaraka* letters [6]. Handwritten Javanese character is an exciting topic to study for scientific purposes and to preserve Indonesian culture. The scope of the research is also very vast, including image processing in Javanese documents, applications changing input written Javanese language text into *Hanacaraka* text and vice versa, Javanese character image classification, and many more.

In image classification, there are some general stages: preprocessing, feature extraction and selection, and classification [7]. There are various method for preprocessing step on handwritten character such as denoising [8]–[13], dilation [14], [15], binarization [16], [17], skeletonization [18]. Some research need feature extraction, such as horizontal and vertical profile image [16], [19], zoning method [11], [18], [20], histogram of oriented gradient (HOG) feature [18], mesh and local line direction (LLD) [17], and fast fourier transform (FFT) [19]. Artificial neural network (ANN) is one powerful machine learning and helpful for classification, clustering, pattern recognition, and prediction [17]. ANN applied in Javanese character problem using backpropagation learning with good performance [10], [15], [21], [22].

Several previous studies focused on handwritten recognition of Javanese script as follows. Fauziah *et al.* [14] obtained 87.5% of accuracy by using dilation, the Otsu method, canny edge, and contour as preprocessing and convolutional neural network (CNN) classifier. Rismiyati *et al.* [18] obtained 88.45% of accuracy by using the HOG feature and support vector machine (SVM). Another Rismiyati *et al.* [23] research did not use any preprocessing technique except grayscaling. They used deep neural network (DNN) and CNN for the recognition step, and the experiment obtained 64.65% and 70.22% on accuracy. Budhi and Adipranat [24] used grayscaling and prefiltering to reduce noise before the segmentation. This study is using image centroid zone-zone centroid zone (ICZ-ZCZ) feature extraction and three classification engines with an accuracy of 3.17% counterpropagation network (CPN), 58.12% evolutionary neural network (ENN) one layer, and 59.31% ENN 2 layers. Widiarti *et al.* [25] used slanting, lowpass filtering, and thinning as preprocessing and the similarity classifier and obtained 79.6% of accuracy average. Wibowo *et al.* [26] using deep learning so is not need either preprocessing or feature extraction technique. Sari *et al.* [27] use median filter and dilation as preprocessing, roundness and eccentricity as features, and K-NN as a classifier. The result gives 87.5% of accuracy and also shows that the preprocessing step, median filter, and dilation significantly improve system accuracy when used together. Sugianela and Suciati [28] use HOG features and multiclass SVM classifier, which shows a higher accuracy value compared to random forest (RF), k-nearest neighbor (kNN), and ANN, 81.3%. Mahastama and Krisnawati [29] proposed optical character recognition (OCR) for Javanese script using projection profile for segmentation, binary image features, and nearest centroid classifier (NCC) for classification. The experiment obtained 60.6% recognition accuracy. Susanto *et al.* [30] use local binary pattern (LBP) as feature extraction and kNN classifier and perform 82.5% of accuracy. Susanto's following research added metrics and eccentricity features. The addition of these two features increased the accuracy of kNN classifier to 92.5% [31]. In another research, Susanto *et al.* [32] use median filter and thresholding as preprocessing and HOG features. Using kNN classier at K=1, the highest accuracy obtained at 98.5%. Rasyidi *et al.* [33] analyze the effect of thinning process as preprocessing step. The result shows that the thinning process did not significantly adjust the accuracy. On the contrary, it decreases from 96.29% to 91.84%. The research uses HOG as feature extraction and RF algorithm as image recognition. Diqi *et al.* [34] compared CNN parameters and pooling filter size and did not implement any preprocessing step. The best accuracy achieves with parameter 5×5 filter average pooling 93%.

This research aimed to analyze preprocessing methods' impact on the handwritten Javanese character dataset. The preprocessing methods are dilation, skeletonization, and noise reduction. To evaluate the effectiveness of each preprocessing method, use the ANN as character recognition.

2. METHOD

This research aims to compare some preprocessing techniques to support the recognition of Javanese handwriting. Figure 1 shows the proposed method. The processing starts with segmentation for region of interest (ROI) extraction, preprocessing using various ways, and finally recognition step to measure the effectiveness of preprocessing step, see Figure 1. As seen in the figure, this research uses a raw preprocessing image without any extracted features as input. After preprocessing, the image matrix, which is the preprocessing result, is directly sent to the classifier.

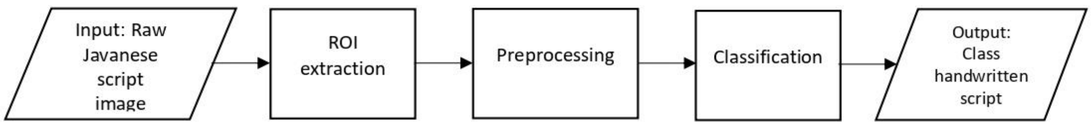

Figure 1. Research method

2.1. *Hanacaraka* letters

Javanese letters, better known as *Hanacaraka*, consist of 20 letters (as seen in Figure 2) from the legend of *Ajisaka*. In the legend, there is a fight between two servants of *Ajisaka*, namely *Dora* and *Sembada*. In the battle, both of them died because they were equally strong, as seen in the meaning of *Hanacaraka* in Figure 2. To commemorate his two servants, *Ajisaka* made the story of the two in alphabetical order [6].

Ha Na Ca Ra Ka =*ono wong loro* (there are two people)

Da Ta Sa Wa La =*podho kerengan* (they both fight)

Pa Dha Ja Ya Nya =*podho joyone* (equally strong)

Ma Ga Ba Tha Nga=*mergo dadi bathang lorone* (hence all of them are dead/both die because they are equally strong)

Figure 2. Javanese letters [6]

2.2. *Hanacaraka* letters dataset

The Javanese script dataset is from the Kaggle database with the name *aksara Jawa*: *aksara Jawa* custom dataset. This dataset consists of 2,154 train data images with different numbers in each class and 480 evaluation data [35]. Table 1 shows distributed data in each class. Each image dataset is an RGB color model and 224×224 pixels in size. Twenty-five example of test data images shows in Figure 3.

Table 1. Number of data train and data test in each class

Class	Number of data train	Number of data test	Class	Number of data train	Number of data test
Ha	102	24	Pa	108	24
Na	108	24	Dha	108	24
Ca	108	24	Ja	108	24
Ra	108	24	Ya	108	24
Ka	108	24	Nya	108	24
Da	108	24	Ma	108	24
Ta	108	24	Ga	108	24
Sa	108	24	Ba	114	24
Wa	108	24	Tha	108	24
La	108	24	Nga	102	24

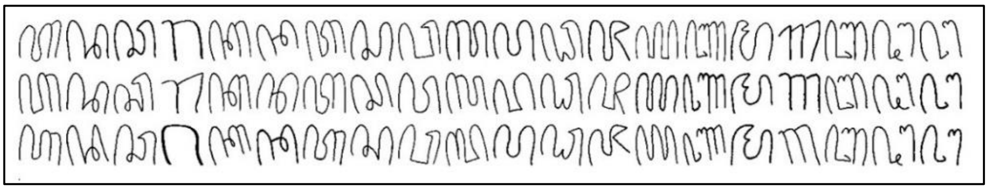

Figure 3. Sample of *Hanacaraka* letters

2.3. Region of interest extraction

The ROI extraction step removes any unnecessary background, so the image is compact with the *Hanacaraka* character object. The shape of the Javanese character is different from the letters of the alphabet, where the form of one Javanese character can consist of two sub-images, such as the letter "nga" in Figure 4. The letter "nga" consists of two sub-image components, which becomes an obstacle in the ROI taking process. Because the letter "*nga*" consists of 2 sub-images, it will produce two convex sub-regions. It is necessary to do a process that helps combine the two sub-images before extracting ROI.

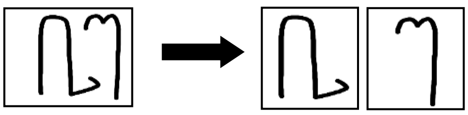

Figure 4. Components of the letter "*nga*"

Figure 5 shows the ROI extraction step. First, the image is inverse, so the object has a white pixel, and the background has a black pixel. Second, closing operation using circle structuring element 15 pixels in diameter. Third, find the extrema point to get the outer corner of an object using regionprops function in MATLAB. There are eight outer corners can be extracted, top-left, top-right, right-top, right-bottom, bottom-right, bottom-left, left-bottom, and left-top. The position of the top-left, for example, is not exactly the top-left of the square bounding box of the object. The fourth step is to solve the problem of finding four square bounding box corners: top-left, top-right, bottom-right, and bottom-left. After finding these four points, the image object *Hanacaraka* character is extracted. Figure 6 is the algorithm to find four corners for ROI extraction of the Javanese character based on the extrema point.

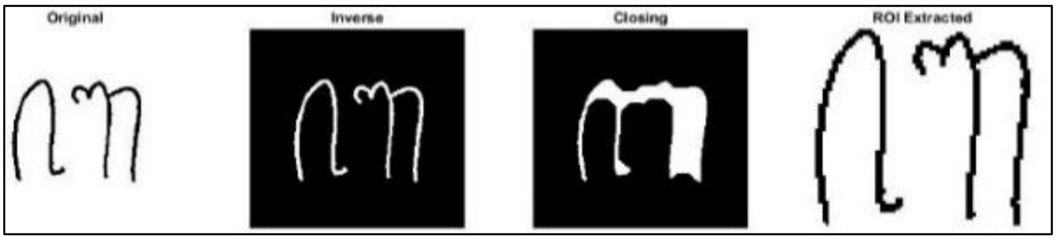

Figure 5. ROI extraction steps

1. Find extrema point
2. Find the minimum row-index of extrema point set and set as minrow
3. Find the maximum row-index of extrema point set and set as maxrow
4. Find the minimum column-index of extrema point set and set as mincol
5. Find the maximum column-index of extrema point set and set as maxcol
6. Get ROI pixel (minrow until maxrow, mincol until maxcol)

Figure 6. ROI extraction algorithm

2.4. Preprocessing

This research uses a combination of preprocessing steps: dilation, skeletonization, and noise reduction. Eight combinations of preprocessing methods are none preprocessing (A), dilation (B), skeletonization (C), noise reduction (D), dilation–skeletonization (E), dilation–noise reduction (F), noise reduction–skeletonization (G), and dilation–noise reduction–skeletonization (H). This research uses two background colors: white and black, with code “w” for white and “b” for black.

2.4.1. Dilation

Dilation is a morphological operation that thickens the object pixels or eliminates small gaps by adding additional pixels around the existing object [36]. Figure 7 shows an example of applying dilation to a binary image with a black background color. The dilation image shows a thickening of the object's pixels according to the size of the structural element.

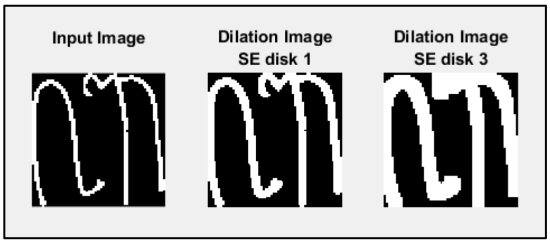

Figure 7. Dilation using disk structural element, (left) original image, (middle) disk=1, (right) disk=3

2.4.2. Skeletonization

The next step, skeletonization, is also a morphological algorithm that aims to obtain a skeleton from the shape of the image object [36]. Skeletonization reduces the object's pixels until it forms a line with one-pixel thickness by successive erosion of A and opening operation, see Figure 8. Like other morphological techniques, skeletonization also uses a kernel to perform its functions.

2.4.3. Noise reduction (denoising)

Denoising aims to remove noise in the image. Noise is a small set of pixels isolated from other parts of the object-denoising method using a filtering method with kernel operators. The application of this method resulted in the blurring/mixing intensity of each pixel to its neighboring pixels. This research uses a Gaussian kernel to eliminate the noise. In the spatial domain, it is usually called filtering or convolution. For $3x3$ kernel (w) in Figure 9, the result output image $g(x,y)$ spatial filtering at point (x,y) in the input image $f(x,y)$ is:

$$g(x,y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} w(s,t) f(x+s, y+t) \quad (1)$$

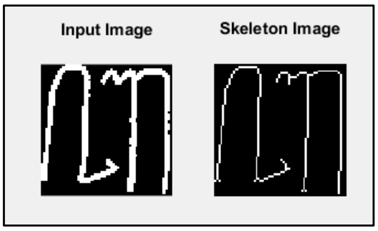

Figure 8. Result of skeletonization

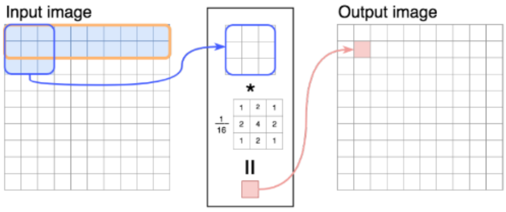

Figure 9. Sliding window spatial filtering [37]

2.5. Artificial neural network classifier

The ANN are the artificial representations of the human brain that simulate the learning process in the human brain. The ANN algorithm consists of interconnected neurons that process input data to output data connected with weight. The ANN learning method, backpropagation, is a supervised, controlled learning algorithm with multiple layers to change the weights associated with neurons in the hidden layer and use the expected output known before [38].

The neural network (NN) structure in this research consists of 20 neurons on the output layer per the number of Javanese characters used: 20 basics (*carakan*) characters. Each neuron has a value of 0 or 1. For example, if the result is character "*ha*," then the first neuron is one while the other neurons are 0. For the second character, "*na*," the second neuron is 1, while the other neurons are 1, and so on. The input neuron use *nxn* image size because there is no feature extraction method, so the raw data is used as input.

This research use ANN to measure the effectiveness of preprocessing step by performance. The experiment uses the various number of hidden layers and hidden nodes. There are three kinds of ANN architecture: three hidden layers, each 32, 64, and 128 nodes; five hidden layers, each 32, 32, 64, 64, and 128 nodes; and seven hidden layers, each 32, 32, 64, 64, 128, 128, and 1,024 nodes. Other ANN parameters are 4,096 input nodes (64×64 matrix) and 20 output nodes representing the class Javanese character.

3. RESULTS AND DISCUSSION

Figure 10 displays the results of applying each preprocessing code. The first line uses a white background input image, while the second uses a black background. The mean squared error (MSE) found the difference between the original image (A) and preprocessed image (B until H). Dw and Db have zero MSE, meaning the noise reduction did not give any change, and the original image does not have any noise. Dilation preprocessing, Bw and Bb, has the biggest MSE because this algorithm makes the thin line thicker. The effect of noise reduction and dilation shows in Fw and Fb, and the noise reduction does not change the image, so only dilation gives an impact, so the MSE is the same as Bw dan Bb. Preprocessing C, E, G, and H involve skeletonization after dilation and/or noise reduction.

Table 2 shows the accuracy performance of each preprocessing combination method in each ANN configuration. In addition, one preprocessing, dilation, gives the best performance in white and black backgrounds. While using two preprocessing steps, the combination of dilation and noise reduction provides the best performance. The highest performance is achieved using three preprocessing on the black background and seven hidden layers of ANN architecture, reaching 98% accuracy. The experiment shows that black background gives good performance overall preprocessing combinations. Otherwise, on white background skeletonization addition makes the performance drop. It can be concluded that the black background gives more performance than the white one due to the skeletonization process, while the skeletonization algorithm calculates the white pixel to find the skeleton of an object, while in this case, the object is in a white pixel.

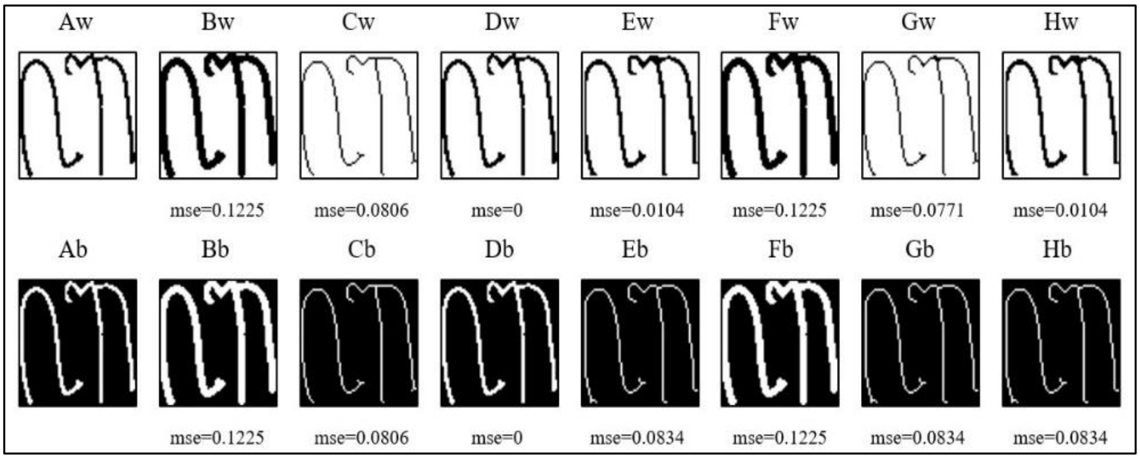

Figure 10. Preprocessing results

Table 2. Testing accuracy

Code	Preprocessing method	Testing accuracy using various number hidden layers and hidden nodes (%)		
		3 hidden layers (32, 64, 128)	5 hidden layers (32, 32, 64, 64, 128)	7 hidden layers (32, 32, 64, 64, 128, 128, 1024)
Aw	White background	69	55	67
Bw	White background and dilation	88	86	87
Cw	White background and skeletonization	5	8	29
Dw	White background and noise reduction	64	60	48
Ew	White background, dilation, and skeletonization	75	69	60
Fw	White background, dilation, and noise reduction	88	80	85
Gw	White background, noise reduction and skeletonization	21	10	27
Hw	White background, dilation, noise reduction and skeletonization	62	65	68
Ab	Black background	95	92	97
Bb	Black background and dilation	95	93	95
Cb	Black background and skeletonization	92	80	94
Db	Black background and noise reduction	95	92	96
Eb	Black background, dilation, and skeletonization	90	82	95
Fb	Black background, dilation, and noise reduction	96	94	95
Gb	Black background, noise reduction and skeletonization	89	78	91
Hb	Black background, dilation, noise reduction and skeletonization	91	86	98
	Average	76	71	77

4. CONCLUSION

In this research, three preprocessing methods (dilation, skeletonization, and noise reduction) are applied to two kinds of background colors (white and black). In the preprocessing experiment by MSE value, noise reduction does not change the image much, so we can remove the existence of noise reduction as preprocessing. Dilation makes the most significant difference on MSE because it thicker the pixel. Otherwise, skeletonization has a little different because it thinner the pixel. The best ANN architecture is seven hidden layers which use 32, 32, 64, 64, 128, 128, and 1,024 nodes on each layer.

The experiment performance shows that the preprocessing methods provide maximum accuracy when used together. A black background image is more suggested than white background because it gives better performance. The noise reduction method can be removed if the image is clear of noise. In future research, other preprocessing methods can be added, and feature extraction also needs to be done to reduce the number of features processed in machine learning. Some feature extraction methods are LBP, gray-level co-occurrence matrix (GLCM), projection profile, and HOG.

ACKNOWLEDGEMENTS

This research was supported by *Lembaga Penelitian dan Pengabdian Masyarakat* (LPPM) Universitas Pembangunan Nasional (UPN) "Veteran" Jawa Timur through the uber publikasi funding scheme. We thank our colleagues who provided insight and expertise that greatly assisted the research.

REFERENCES

[1] A. A. Hidayat, K. Purwandari, T. W. Cenggoro, and B. Pardamean, "A Convolutional Neural Network-based Ancient Sundanese Character Classifier with Data Augmentation," in *Procedia Computer Science*, vol. 179, pp. 195–201, Jan. 2021, doi: 10.1016/j.procs.2020.12.025.

[2] A. Imron, "The development of Iqra' Lampung script teaching materials for primary school levels in Bandar Lampung city," *International Journal of Educational Studies in Social Sciences*, vol. 1, no. 1, pp. 38–43, 2021, [Online]. Available: https://ijesss.com/journal/article/view/5/5.

[3] A. Hidayat, "Revitalization of ancient Indonesian characters and the maintenance efforts in past 10 years," *LADU: Journal of Languages and Education*, vol. 1, no. 4, pp. 179–186, May 2021, doi: 10.56724/ladu.v1i4.69.

[4] Y. S. Baso and A. Agussalim, "Computerization of Local Language Characters An Innovative Model for Language Maintenance in South Sulawesi, Indonesia," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 12, no. 12, pp. 76–84, 2021, doi: 10.14569/IJACSA.2021.0121211.

[5] E. M. Zamzami, S. Hayanti, and E. B. Nababan, "Diagonal Based Feature Extraction and Backpropagation Neural Network in Handwritten Batak Toba Characters Recognition," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 6, no. 2, pp. 117–126, May 2021, doi: 10.22219/kinetik.v6i2.1212.

[6] F. Sari, Mr. Supana, and S. Suwandi, "Problem based learning model assisted edmodo," in *Proceedings of the 1st International Conference on Education Innovation (ICEI 2017)*, Feb. 2018, pp. 112–116. doi: 10.2991/icei-17.2018.30.

[7] S. V. S. Prasad, T. S. Savithri, and I. V. M. Krishna, "Techniques in image classification; A survey," *Global Journal of Researches in Engineering: Electrical and Electronics Engineering*, vol. 15, no. 6, pp. 17–32, 2015.

[8] G. A. Robby, A. Tandra, I. Susanto, J. Harefa, and A. Chowanda, "Implementation of optical character recognition using tesseract with the javanese script target in android application," *Procedia Computer Science*, vol. 157, pp. 499–505, 2019, doi: 10.1016/j.procs.2019.09.006.

[9] L. R. Mursari and A. Wibowo, "The effectiveness of image preprocessing on digital handwritten scripts recognition with the implementation of OCR Tesseract," *Computer Engineering and Applications Journal*, vol. 10, no. 3, pp. 177–186, Oct. 2021, doi: 10.18495/comengapp.v10i3.386.

[10] A. N. Handayani, H. W. Herwanto, K. L. Chandrika, and K. Arai, "Recognition of handwritten Javanese script using backpropagation with zoning feature extraction," *Knowledge Engineering and Data Science*, vol. 4, no. 2, pp. 117–127, Dec. 2021, doi: 10.17977/um018v4i22021p117-127.

[11] H. W. Herwanto, A. N. Handayani, K. L. Chandrika, and A. P. Wibawa, "Zoning feature extraction for handwritten Javanese character recognition," in *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, Oct. 2019, pp. 264–268, doi: 10.1109/ICEEIE47180.2019.8981462.

[12] N. D. Priandani and F. Utaminingrum, "E-evaluation measurement for Javanese script handwriting studies," in *The 6th Annual Basic Science International Conference*, 2016, pp. 73–76.

[13] F. Damayanti, Y. K. Suprapto, and E. M. Yuniarno, "Segmentation of Javanese character in ancient manuscript using connected component labeling," in *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, Nov. 2020, pp. 412–417, doi: 10.1109/CENIM51130.2020.9297954.

[14] Y. Fauziah, K. Aprilianta, and H. C. Rustamaji, "Convolutional neural network method for classification of syllables in Javanese script," *International Journal of Artificial Intelligence & Robotics (IJAIR)*, vol. 3, no. 2, pp. 80–89, Nov. 2021, doi: 10.25139/ijair.v3i2.4395.

[15] A. Setiawan, A. S. Prabowo, and E. Y. Puspaningrum, "Handwriting character recognition Javanese letters based on artificial neural network text mining and text information retrieval view project artificial intelligence view project handwriting character recognition Javanese letters based on artificial neura," *Network Security and Information System (IJCONSIST)*, vol. 1, no. 1, pp. 39–42, 2019, doi: 10.33005/ijconsist.v1i1.12.

[16] A. R. Widiarti and P. N. Wastu, "Javanese character recognition using hidden Markov model," *International Journal of Computer and Information Engineering*, vol. 3, no. 9, pp. 251–261, 2009.

[17] B. Karundeng, K. I. Eng, and A. S. Nugroho, "An evaluation of feature extraction algorithms for automatic language transcription system for ancient handwriting Javanese manuscripts," *International Seminar on Industrial Engineering and Management*, pp. 659–665, 2012.

[18] Rismiyati, Khadijah, and D. E. Riyanto, "HOG and zone base features for handwritten Javanese character classification," in *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)*, Oct. 2018, pp. 1–5, doi: 10.1109/ICICOS.2018.8621781.

[19] S. Wardoyo, K. W., A. S. Pramudyo, S. Suhendar, and S. Hidayat, "The development of Javanese language teaching materials through introduction of Java scripts using artificial neural network," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 4, pp. 1697–1703, Aug. 2018, doi: 10.12928/telkomnika.v16i4.8465.

[20] L. D. Krisnawati and A. W. Mahastama, "Building classifier models for on-off Javanese character recognition," in *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*, Dec. 2019, pp. 25–34. doi: 10.1145/3366030.3366050.

[21] I. Prihandi, I. Ranggadara, S. Dwiasnati, Y. S. Sari, and Suhendra, "Implementation of backpropagation method for identified Javanese scripts," *Journal of Physics: Conference Series*, vol. 1477, no. 3, pp. 1–6, Mar. 2020, doi: 10.1088/1742-6596/1477/3/032020.

[22] G. S. Budhi and R. Adipranata, "Handwritten Javanese character recognition using several artificial neural network methods," *Journal of ICT Research and Applications*, vol. 8, no. 3, pp. 195–212, Aug. 2015, doi: 10.5614/itbj.ict.res.appl.2015.8.3.2.

[23] Rismiyati, Khadijah, and A. Nurhadiyatna, "Deep learning for handwritten Javanese character recognition," in *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*, Nov. 2017, pp. 59–64, doi: 10.1109/ICICOS.2017.8276338.

[24] G. S. Budhi and R. Adipranata, "Comparison of bidirectional associative memory, counterpropagation and evolutionary neural network for Java characters recognition," in *2014 International Conference of Advanced Informatics: Concept, Theory and Application (ICAICTA)*, Aug. 2014, pp. 7–10, doi: 10.1109/ICAICTA.2014.7005906.

[25] A. R. Widiarti, A. Harjoko, Marsono, and S. Hartati, "The model and implementation of Javanese script image transliteration," in *2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT)*, Sep. 2017, pp. 51–57, doi: 10.1109/ICSIIT.2017.17.

[26] M. A. Wibowo, M. Soleh, W. Pradani, A. N. Hidayanto, and A. M. Arymurthy, "Handwritten Javanese character recognition using descriminative deep learning technique," in *2017 2nd International conferences on Information Technology, Information

Systems and Electrical Engineering (ICITISEE), Nov. 2017, pp. 325–330, doi: 10.1109/ICITISEE.2017.8285521.
[27] C. A. Sari, M. W. Kuncoro, D. R. I. M. Setiadi, and E. H. Rachmawanto, "Roundness and eccentricity feature extraction for Javanese handwritten character recognition based on K-nearest neighbor," in *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Nov. 2018, pp. 5–10, doi: 10.1109/ISRITI.2018.8864252.
[28] Y. Sugianela and N. Suciati, "Javanese document image recognition using multiclass support vector machine," *CommIT (Communication and Information Technology) Journal*, vol. 13, no. 1, pp. 25–30, May 2019, doi: 10.21512/commit.v13i1.5330.
[29] A. W. Mahastama and L. D. Krisnawati, "Optical character recognition for printed Javanese script using projection profile segmentation and nearest centroid classifier," in *2020 Asia Conference on Computers and Communications (ACCC)*, Sep. 2020, pp. 52–56, doi: 10.1109/ACCC51160.2020.9347895.
[30] A. Susanto, D. Sinaga, C. A. Sari, E. H. Rachmawanto, and D. R. I. M. Setiadi, "A high performace of local binary pattern on classify Javanese character classification," *Scientific Journal of Informatics*, vol. 5, no. 1, pp. 1–8, May 2018, doi: 10.15294/sji.v5i1.14017.
[31] A. Susanto, I. U. W. Mulyono, C. A. Sari, E. H. Rachmawanto, and D. R. I. M. Setiadi, "Javanese script recognition based on metric, eccentricity and local binary pattern," in *2021 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2021, pp. 118–121, doi: 10.1109/iSemantic52711.2021.9573232.
[32] A. Susanto, C. A. Sari, I. U. W. Mulyono, and M. Doheir, "Histogram of gradient in k-nearest neighbor for Javanese alphabet classification," *Scientific Journal of Informatics*, vol. 8, no. 2, pp. 289–296, Nov. 2021, doi: 10.15294/sji.v8i2.30788.
[33] 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, Jun. 2021, doi: 10.11591/eei.v10i3.3036.
[34] M. Diqi, R. N. Wijaya, and M. F. Muhdalifah, "Pooling comparison in CNN architecture for Javanese script classification," *International Journal of Informatics and Computation*, vol. 3, no. 2, pp. 15–22, Jan. 2022, doi: 10.35842/ijicom.v3i2.30.
[35] "Aksara Jawa | Kaggle." https://www.kaggle.com/datasets/phiard/aksara-jawa (accessed May. 11, 2022).
[36] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 2nd ed. New Jersey: Prentice Hall, 2002.
[37] O. Shipitko and A. Grigoryev, "Gaussian filtering for FPGA based image processing with high-level synthesis tools," in *Proceedings of the IV International Conference on Information Technology and Nanotechnology, Sarma, Russia*, 2018, pp. 1–6.
[38] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, pp. 1–41, Nov. 2018, doi: 10.1016/j.heliyon.2018.e00938.

BIOGRAPHIES OF AUTHORS

Fetty Tri Anggraeny is an Assistant Professor at the Department of Informatics, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia and has been served as a lecturer since 2006. She received the bachelor degree in Computer Science from the Department of Informatics, Institut Teknologi Sepuluh Nopember (ITS), Surabaya-Indonesia in 2005. Then, she received her master degree in Computer Science from the Department of Informatics, Institut Teknologi Sepuluh Nopember (ITS), Surabaya-Indonesia in 2012. Her research interests are primarily in the area of artificial intelligence, computer vision, and digital image processing. She has been the author/co-author of over 70 research publications and three books. She can be contacted at email: fettyanggraeny.if@upnjatim.ac.id.

Yisti Vita Via received Bachelor of Information Technology from Politeknik Elektronika Negeri Surabaya (PENS), Indonesia in 2008. She then received Master of Informatics Enginering from Institut Teknologi Sepuluh Nopember (ITS) Surabaya, Indonesia in 2012. Currently, she is a lecturer at the Department of Informatics, Faculty of Computer Science, Universitas Pembangunan Nasional Veteran Jawa Timur Surabaya, Indonesia. Her research interest includes the development of intelligent systems by applying nature-inspired algorithms and optimization algorithms. She can be contacted at email: yistivia.if@upnjatim.ac.id.

Retno Mumpuni received the bachelor degree in Computer Science from the Department of Informatics, Institut Teknologi Sepuluh Nopember (ITS), Surabaya-Indonesia in 2011. Then, she received her master degree from Department Computer Science and Information Engineering in National Taiwan University of Science and Technology in 2015. Currently she is served as a lecturer at Universitas Pembangunan Nasional "Veteran" Jawa Timur since 2017. Her research interests include software engineering, database, and data engineering. She can be contacted at email: retnomumpuni.if@upnjatim.ac.id.