Bulletin of Electrical Engineering and Informatics
Vol. 10, No. 3, June 2021, pp. 1308~1315
ISSN: 2302-9285, DOI: 10.11591/eei.v10i3.3036

Classification of handwritten Javanese script using random forest algorithm

Mohammad Arif Rasyidi, Taufiqotul Bariyah, Yohanes Indra Riskajaya, Ayunda Dwita Septyani
Department of Informatics, Universitas Internasional Semen Indonesia, Indonesia

Article Info

Article history:

Received Nov 10, 2020
Revised Mar 20, 2021
Accepted Apr 23, 2021

Keywords:

Handwriting recognition
Javanese script
Pattern recognition
Random forest

ABSTRACT

The ability to read and write Javanese scripts is one of the most important competencies for students to have in order to preserve the Javanese language as one of the Indonesian cultures. In this study, we developed a predictive model for 20 Javanese characters using the random forest algorithm as the basis for developing Javanese script learning media for students. In building the model, we used an extensive handwritten image dataset and experimented with several different preprocessing methods, including image conversion to black-and-white, cropping, resizing, thinning, and feature extraction using histogram of oriented gradients. From the experiment, it can be seen that the resulting random forest model is able to classify Javanese characters very accurately with accuracy, precision, and recall of 97.7%.

Corresponding Author:

Mohammad Arif Rasyidi
Department of Informatics
Universitas Internasional Semen Indonesia
Jl. Veteran, Gresik, Jawa Timur 61122, Indonesia
Email: mohammad.rasyidi@uisi.ac.id

1. INTRODUCTION

Javanese language is one of the oldest cultural heritage in Indonesia. Javanese was born as one of the languages that characterizes Indonesia as a nation. It is a language that is widely used by the world community. In 2007, it was noted that Javanese was spoken by 82 million or 1.25% of the world's population [1]. However, by 2015 this number has actually decreased to 68.2 million [2] which is clearly not proportional to the increase in the world's population. With the reduced use of language and script, it will become a threat to the loss of Indonesian culture.

One of the things that can be done to overcome the problem is to intensively introduce the language to students as the next generation and speakers of the language. Currently, Javanese is taught as local content at the primary and secondary education levels in some regions [3]. The basic competencies taught in Javanese subjects are *wayang*, *tembang*, *geguritan*, fairy tales, traditional games, and Javanese script. Among these, Javanese script is one of the competencies that is very important for students to learn. However, because Javanese is a local content subject, the teaching portion is quite small compared to other compulsory subjects such as mathematics and English. Furthermore, not all schools teach it to their students. With the small amount of teaching time and the variety of other materials in Javanese lessons, students may experience difficulties memorizing and writing Javanese letters or characters. Coupled with the variety and complexity of these characters, Javanese script reading and writing competencies are often not optimally conveyed to students.

To overcome these problems, some media to support Javanese script learning need to be developed. One of the development concepts that can be done is by building a system that asks students to write

Javanese characters according to the questions given. The system will then work based on the image classification technique where the handwritten results of the students will be identified using the previously developed handwriting prediction model. If the results match the expected answers, the students will get a score and can continue to the next question. Meanwhile, if the results are wrong or not suitable, then the students will be asked to start over. It is hoped that with the developed system, students will be able to learn independently so that the competence of writing Javanese characters can be better mastered.

It can be seen from the description that the most crucial component in the learning system being developed is the classification model. If the performance of the classification model is not good, unwanted things will occur, such as student answers being classified as wrong even though they are true or vice versa, wrong answers are considered as correct answers. This of course will cause the low quality of the system being developed.

In previous studies, the problem of handwriting classification has been widely researched. The most popular problem is the MNIST database [4], which is the handwriting recognition of digits (numbers 0 to 9). This problem has been researched and solved by several methods, including linear classifier [4], K-nearest neighbors (KNN) [5], support vector machine (SVM) [6], artificial neural network (ANN) [7], and convolutional neural network (CNN) [8]. The best model for current MNIST digit classification can yield an accuracy of more than 99% or an error of less than 1% [8].

Apart from MNIST, handwriting classification in various languages and scripts has also been widely explored. For example, [9] described the handwriting recognition mechanism used by Google to identify handwriting in 22 scripts with an average error rate of below 10% while [10] developed a handwritten alphabet recognition application using ANN and produced an accuracy of 86.535%. In the field of Javanese script recognition itself, [11] have implemented a combination of SVM and directional element feature to produce an accuracy of 93.6%. However, the research conducted in [11] only used a very limited amount of data (50 training data and 10 test data for each character) so that its application needs to be studied again for a larger and more varied amount of data. In another study, the classification of Javanese characters was carried out using ANN [12]. However, the performance of the resulting model is still unsatisfactory with an average accuracy of only 73%. Deep learning approaches, such as CNN, which is well-known for its image classification capabilities, e.g. in [13]-[15], have also been commonly used in the field of Javanese handwriting recognition, for example in [16]-[18]. Their applications, however, must be reviewed due to limited data and unsatisfactory performance. Finally, several feature extraction methods have also been extensively explored in [19]-[21]. The results show that by employing feature extraction, some traditional machine learning methods such as KNN are able to produce fairly good accuracy of more than 80%.

In this study, we propose the application of the random forest algorithm [22] to solve the Javanese script classification problem. The random forest algorithm is an ensemble learning application that combines several Decision Tree models in order to make predictions. This algorithm has been widely implemented and produces excellent performance in various related studies, including handwriting recognition with an accuracy rate above 90% [23]. Our contributions are as follows. To the best of our knowledge, we are the first to focus on implementing the random forest algorithm for classifying Javanese characters. There has been previous research comparing the performance of the random forest algorithm for Javanese script classification [24]. However, that study focused more on the application of SVM for its classification method. Another contribution is that in this study, we also experimented with a fairly large amount of data and extensively compared several data preprocessing schemes to find out which one is more efficient to use in Javanese script classification problems.

2. RESEARCH METHODOLOGY

2.1. Data collection

Our research methodology is illustrated in Figure 1. The data used in this research are Javanese script handwritten image data. We use the 20 characters of the *Nglegena* character set (*ha, na, ca, ra, ka, da, ta, sa, wa, la, pa, dha, ja, ya, nya, ma, ga, ba, tha, nga*) shown in Figure 2. To collect this data, we asked our respondents to copy the 20 Javanese characters in a form shown in Figure 3 which were then scanned and cropped for each character. From this process, 6000 images were collected (300 images for each character). The data is further divided into 2 datasets: training data and test data with a ratio of 7:3.

2.2. Data augmentation

Machine learning requires large amounts of data to improve the quality of the resulting model and avoid overfitting. In this research, we use image augmentation to increase the variety of our data. Random rotations and shears are performed to augment each image in both training data and test data 4 times as

illustrated in Figure 4. From this process combined with the original image data, we obtained training and test data consisting of 21000 and 9000 images, respectively.

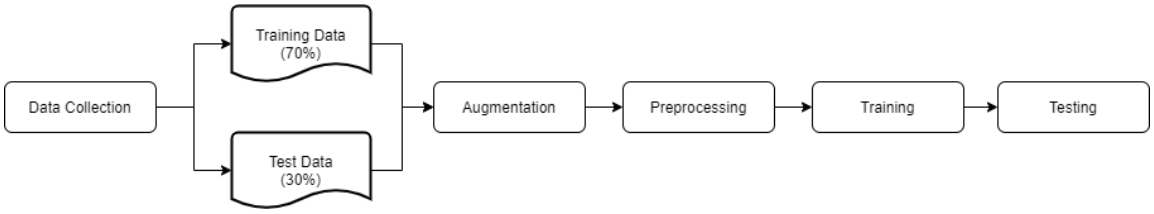


Figure 1. Research methodology

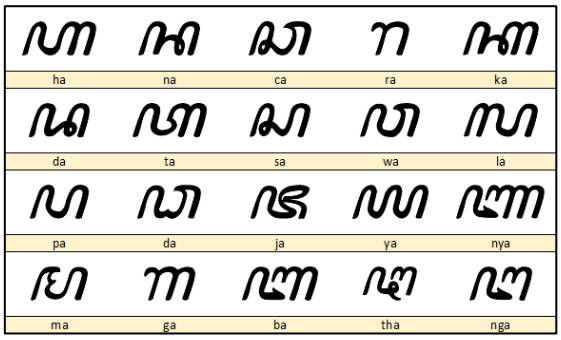
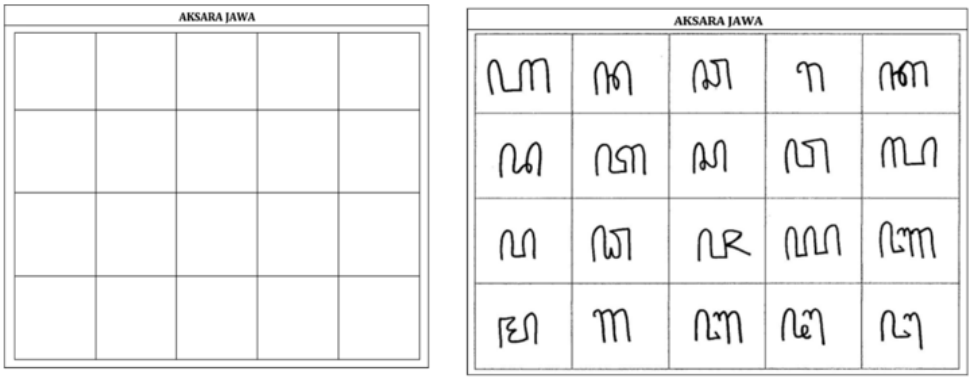


Figure 2. *Nglegena* Javanese characters



(a) (b)

Figure 3. Example of forms that have not, (a) And have, (b) Been filled in

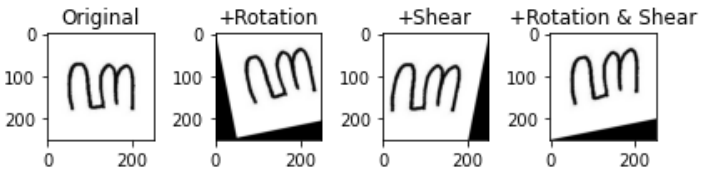


Figure 4. Illustration of data augmentation process

2.3. Data preprocessing

Image data that have been collected in the previous step are then preprocessed in several stages, namely converting them into black-and-white images, cropping, and resizing. In the first stage, the data will be

converted into black and white images. This step is carried out according to the pseudocode shown in Figure 5. The resulting images are then cropped to remove the empty space around the character. This cropping process is simply implemented using pseudocode in Figure 6. Finally, the resulting images will be resized into 32x32 pixels. An example of the results of these three stages is shown in Figure 7.

In addition to using these three stages, in this study we also tried several additional scenarios, namely by using the thinning process to attenuate the lines of the characters as shown in Figure 7. In addition, we also experimented with the feature extraction process using the histogram of oriented gradients (HOG) [25]. Thus, there are 4 variations of data that will be used and compared in the training and testing process, namely data without thinning and HOG, data with thinning only, data with HOG only, and data with both thinning and HOG.

```
threshold = 200
for x = 1 to imgWidth:
    for y = 1 to imgHeight:
        if img[x, y] < threshold:
            img[x, y] = 0
        else
            img[x, y] = 255
```

Figure 5. Pseudocode of the image conversion process into black and white

```
minX = imgWidth
maxX = 0
minY = imgHeight
maxY = 0

for x = 1 to imgWidth:
    for y = 1 to imgHeight:
        if img[x, y] = 255:
            if (x < minX) minX = x
            if (x > maxX) maxX = x
            if (y < minY) minY = y
            if (y > maxY) maxY = y

img = img[minX:maxX, minY:maxY]
```

Figure 6. Pseudocode of the cropping process

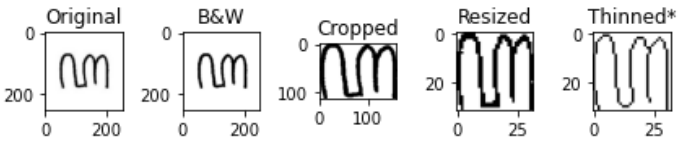


Figure 7. Sample output of each stage in the preprocessing. Note that the thinning process is optional

2.4. Training

To train our random forest models, Grid Search with 3-fold cross validation is employed to find the best combination of parameters for the model. The explored parameters are shown in Table 1 with a total of 20 combinations.

Table 1. Parameters of grid search

Parameter	Values
Impurity measure	Gini, entropy
Number of trees	200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000

2.5. Testing

The best model found from the training process using the grid search will be tested using the previously prepared test data. From the test results, accuracy, precision, and recall will be calculated and reported based on the following formulas;

$$\text{accuracy} = \frac{\sum_{i=1}^{k} \text{TP}_i}{\text{n}}$$

$$precision = \frac{\sum_{i=1}^{k} \frac{TP_i}{TP_i + FP_i}}{k}$$

$$recall = \frac{\sum_{i=1}^{k} \frac{TP_i}{TP_i + FN_i}}{k}$$

where:
TP=true positive
FN=false negative
FP=false positive
n=number of data
k=number of classes

3. RESULTS AND DISCUSSION

Table 2 shows the performance of the random forest model generated using the four types of data used. It can be seen that the model generated from data without the thinning and HOG processes produces the best performance compared to other models when viewed from accuracy, precision and recall aspects. This result is quite surprising, considering that the thinning process produces images that are quite easily recognized by the human eye and HOG is a feature extraction method that is widely used in image classification problems. However, in this Javanese script handwriting recognition problem, both actually produce slightly worse performance when compared to data without any additional treatment. This may be due to the small size of the image used so that any additional treatment will slightly reduce the information contained in the images that may be needed in the recognition process.

In terms of the best parameters for model development, almost all models produce the best parameters using gini as the impurity measure except the model from data with additional thinning treatment and feature extraction with HOG, which uses entropy as the impurity measure, which is shown in Table 3. It can also be seen that all models tend to require a fairly large number of trees, which makes sense considering that as the number of trees in the random forest model increases, the tendency of the model to overfit tends to decrease.

Table 2. The performance of the resulting models

Data	Accuracy	Precision	Recall
No Thin, No HOG	**0.9777**	**0.9778**	**0.9777**
Thin	0.9087	0.9101	0.9087
HOG	0.9629	0.9631	0.9629
Thin+HOG	0.9184	0.9185	0.9185

Table 3. The best parameters

Data	Impurity Measure	Number of Trees
No Thin, No HOG	Gini	1800
Thin	Gini	2000
HOG	Gini	2000
Thin+HOG	Entropy	2000

Next, we look at the confusion matrix of the model from the data without thinning and HOG feature extraction to see what characters are most often incorrectly predicted. It can be seen in Figure 8 that the character that is most often wrongly predicted is the character *tha* which is predicted as *nga* 20 times. As can be seen in Figure 9, the *tha* character is indeed very similar to the *nga* character so that the model has difficulty distinguishing the two. To deal with this problem, in future studies, the quality and quantity of data needs to be improved.

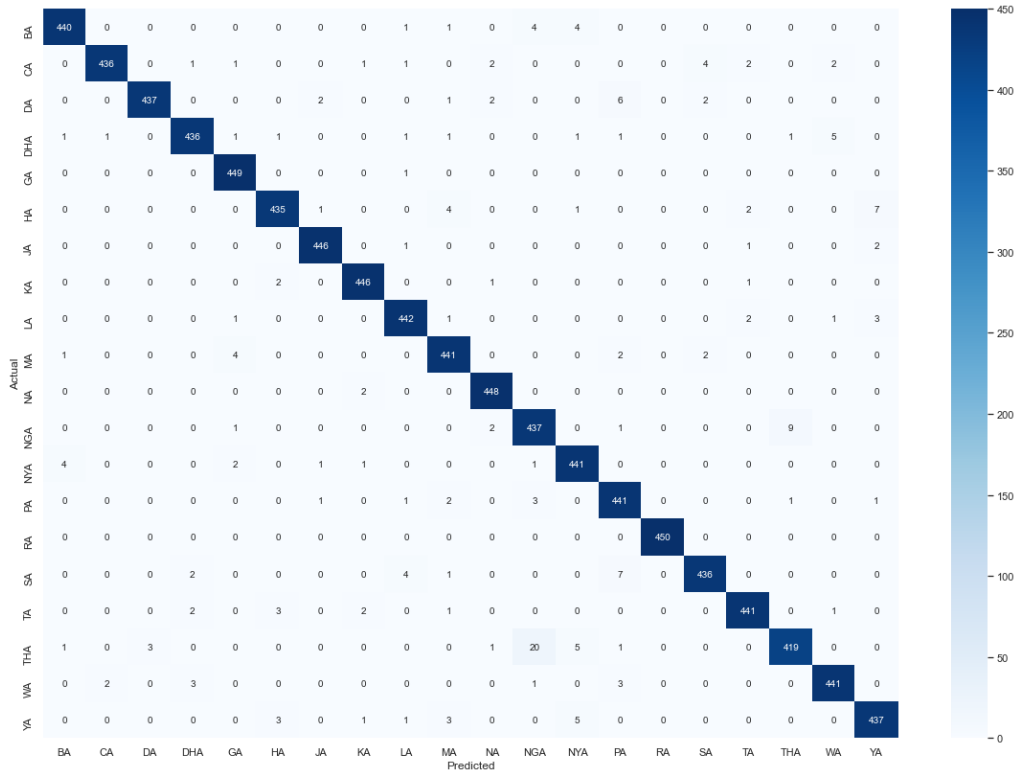


Figure 8. The confusion matrix of the model produced from data without thinning and HOG feature extraction

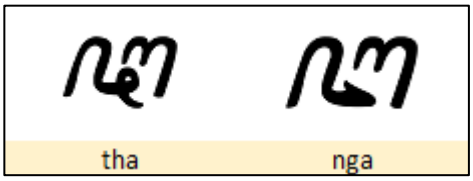


Figure 9. The most confused characters: *tha* and *nga*

4. CONCLUSION

From the results of this study, we have seen that the Random Forest algorithm can be applied in identifying handwritten Javanese characters and shows very good performance. The best performance is obtained through preprocessing by converting the images into black-and-white, cropping, and resizing. Additional steps, namely thinning and feature extraction with HOG do not result in better performance. This is presumably because the image size is small enough so that the additional step may actually reduce the useful information in the image, which is needed in the character recognition process. In future research, we would like to compare the performance of traditional machine learning algorithms such as random forest, SVM, and KNN to the state-of-the-art method, that is deep learning with CNN on the same problem. CNN has been recognized as one of the most powerful methods for image classification. It would be interesting to see if the performance of a model generated using CNN is significantly better than that of a typical machine learning model on this particular problem.

ACKNOWLEDGEMENTS

This research is supported by Kementerian Riset dan Teknologi/Badan Riset dan Inovasi Nasional Indonesia.

REFERENCES

[1] M. Parkvall, "Världens 100 största språk 2007' (The World's 100 Largest Languages in 2007)," in Nationalencyklopedin, 2007.

[2] Ethnologue, "Javanese," 2019. [Online]. Available: https://www.ethnologue.com/language/jav (accessed Aug. 20, 2019).

[3] P. K. B. Kemendiknas, "Pengembangan Pendidikan Budaya dan Karakter Bangsa," *Jakarta: Balitbang: Kemendiknas*, 2010.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998, doi: 10.1109/5.726791.

[5] D. Keysers, T. Deselaers, C. Gollan, and H. Ney, "Deformation models for image recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1422-1435, 2007, doi: 10.1109/TPAMI.2007.1153.

[6] D. Decoste and B. Schölkopf, "Training invariant support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 161-190, 2002, doi: 10.1023/A:1012454411458.

[7] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Computation*, vol. 22, no. 12, pp. 3207-3220, 2010, doi: 10.1162/NECO_a_00052.

[8] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification Supplementary Online Material," 2012 *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012, pp. 3642-3649, doi: 10.1109/CVPR.2012.6248110.

[9] D. Keysers, T. Deselaers, H. A. Rowley, L. L. Wang, and V. Carbune, "Multi-Language Online Handwriting Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1180-1194, 2017, doi: 10.1109/TPAMI.2016.2572693.

[10] H. Masrani, Ilhamsyah, and I. Ruslianto, "Aplikasi Pengenalan Pola Pada Huruf Tulisan Tangan Menggunakan Jaringan Saraf Tiruan Dengan Metode Ekstraksi Fitur Geometri," *Coding Jurnal Komputer dan Aplikasi*, vol. 6, no. 2, pp. 69-78, 2018, doi: 10.26418/coding.v6i2.26674.

[11] A. H. Nurul, M. D. Sulistiyo, and R. N. Dayawati, "Pengenalan Aksara Jawa Tulisan Tangan Menggunakan Directional Element Feature Dan Multi Class Support Vector Machine," *Konferensi Nasional Teknologi Informasi dan Aplikasinya*, vol. 3, pp. A13-A22, 2014.

[12] 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, 2015, doi: 10.5614/itbj.ict.res.appl.2015.8.3.2.

[13] M. A. Rasyidi and T. Bariyah, "Batik pattern recognition using convolutional neural network*," Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 9, no. 4, pp. 1430-1437, 2020, doi: 10.11591/eei.v9i4.2385.

[14] Y. Seo and K. shik Shin, "Hierarchical convolutional neural networks for fashion image classification," *Expert Systems with Applications*, vol. 116, pp. 328-339, 2019, doi: 10.1016/j.eswa.2018.09.022.

[15] L. Fang, Y. Jin, L. Huang, S. Guo, G. Zhao, and X. Chen, "Iterative fusion convolutional neural networks for classification of optical coherence tomography images," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 327-333, 2019, doi: 10.1016/j.jvcir.2019.01.022.

[16] C. K. Dewa, A. L. Fadhilah, and A. Afiahayati, "Convolutional Neural Networks for Handwritten Javanese Character Recognition," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 12, no. 1, pp. 83-94, 2018, doi: 10.22146/ijccs.31144.

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

[18] M. A. Wibowo, M. Soleh, W. Pradani, A. N. Hidayanto, and A. M. Arymurthy, "Handwritten javanese character recognition using descriminative deep learning technique," *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, 2017, pp. 325-330, doi: 10.1109/ICITISEE.2017.8285521.

[19] 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," *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Yogyakarta, Indonesia, 2018, pp. 5-10, doi: 10.1109/ISRITI.2018.8864252.

[20] H. W. Herwanto, A. N. Handayani, K. L. Chandrika, and A. P. Wibawa, "Zoning Feature Extraction for Handwritten Javanese Character Recognition," *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, Denpasar, Indonesia, 2019, pp. 264-268, doi: 10.1109/ICEEIE47180.2019.8981462.

[21] Rismiyati, Khadijah, and D. E. Riyanto, "HOG and Zone Base Features for Handwritten Javanese Character Classification," *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)*, Semarang, Indonesia, 2018, pp. 1-5, doi: 10.1109/ICICOS.2018.8621781.

[22] T. K. Ho, "Random decision forests," *Proceedings of 3rd International Conference on Document Analysis and Recognition*, Montreal, QC, Canada, 1995, pp. 278-282, vol. 1, doi: 10.1109/ICDAR.1995.598994.

[23] S. Bernard, L. Heutte, and S. Adam, "Using random forests for handwritten digit recognition," *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Curitiba, Brazil, 2007, pp. 1043-1047, doi: 10.1109/ICDAR.2007.4377074.

[24] 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, 2019, doi: 10.21512/commit.v13i1.5330.
[25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 886-893, vol. 1, doi: 10.1109/CVPR.2005.177.

BIOGRAPHIES OF AUTHORS

Mohammad Arif Rasyidi earned a bachelor's degree in Information Systems from Sepuluh Nopember Institute of Technology, Indonesia in 2012 and a master's degree in Electrical and Computer Engineering from Pusan National University, Korea in 2015. His research interests include Machine Learning and Evolutionary Computation.

Taufiqotul Bariyah earned a bachelor's degree in Informatics from Sepuluh Nopember Institute of Technology, Indonesia in 2013 and a master's degree in Information Management at the National Taiwan University of Science and Technology in 2017. Her research interests include Transportation Management, Logistics, and Optimization.

Yohanes Indra Riskajaya received his bachelor and master's degree in Informatics from Sepuluh Nopember Institute of Technology, Indonesia in 2009 and 2015 respectively. His research field includes network-based computing.

Ayunda Dwita Septyani graduated from Universitas Internasional Semen Indonesia with a bachelor's degree in Informatics in 2019. During college, she did an internship at PT Ume Persada Indonesia, where she created an HR management information system for the company. She is currently searching for a permanent position in the IT industry.