ICIC Express Letters
Volume **17**, Number **4**, April **2023**

ICIC International ©2023 ISSN 1881-803X
pp. **409–417**

COMPARISON STUDY OF IMAGE AUGMENTATION ON MODIFIED CNN ARCHITECTURE FOR INDONESIAN LASEM-BATIK'S MOTIFS

Ida Bagus Kerthyayana Manuaba[1,*] and Vera Jenny Basiroen[2]

[1]Computer Science Department, Faculty of Computing and Media
[2]Graphic Design & New Media Program, Visual Communication Design Department
BINUS Northumbria School of Design
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
nonjane@binus.edu
*Corresponding author: imanuaba@binus.edu

Received April 2022; accepted June 2022

Abstract. *Batik is a fabric art originally from Indonesia which is a technique that utilizes hot wax to draw a specific design and motif on a fabric. As part of many efforts in preserving batik in Indonesia, this research study undertakes a technology approach to digitize batik pattern and motifs and learn how to identify the patterns for future reference. The research study focuses on using one of Indonesia's diverse batik motifs, known as Batik Lasem, that are well-known for their distinctive pattern. In this paper, we manually collected around five thousand unique samples of Lasem-batik motifs and used them as machine learning (ML) data using modified convolutional neural network (CNN) architecture with Tensorflow 2.0 and Keras library. This paper focuses on the implementation of data augmentation to explore how well it could improve our ML model in training and validating the data. By evaluating accuracy, loss, and mean squared error (MSE) scores, we tried to investigate the performance of our developed ML model and focused on the data augmentation implementation to investigate any improvements in the process of training and validating ML data. Further analysis by using precision, recall, and F1-scores was also conducted to analyze which batik motifs fit better in our developed ML model.*
Keywords: Lasem-batik, Image augmentation, Modified CNN, Machine learning, Performance analysis

1. **Introduction.** As a handmade fabric art originally from Indonesia, the word 'batik' is also known as a drawing technique on a fabric using dots and lines of hot wax. Based on the root name, batik comes from the Javanese words '*amba*' which means 'to write' and '*titik*' which means 'dot' [1].

As one of the national cultural signifiers, the diversity of batik can be easily found in various regions in Indonesia, especially on the island of Java [2]. By looking at the motifs and patterns, people can easily distinguish the origin of the batik.

One well-known batik in Indonesia is known as 'Batik Lasem', which originated from Lasem, in the Rembang area of Central Java. This location is known as a melting pot of Javanese and Chinese cultures where they influenced a lot of the characteristics of the pattern and motifs of Lasem-batik [3]. Some of the unique motifs of Lasem-batik are *Gunung Ringgit*, *Watu Pecah (Kricak)*, *Latohan*, *Nyuk Pitu* and *Seritan* [4].

Preservation of batik in Indonesia is important to be carried out so as not to be eroded by the march of time [5]. The objective of this research study is to utilize advanced technology to digitize one set of Indonesian batik motifs, known as Batik Lasem. Hence, in this paper we are discussing more focus on the development of a machine learning (ML)

DOI: 10.24507/icicel.17.04.409

model with modified convolutional neural network (CNN) architecture that could be used to learn, identify, and predict the Lasem-batik motifs for future reference for the younger generations.

The CNN is one of artificial intelligent algorithms which has been widely used to work with an image and assign learnable weights and biases to specific aspects in the image that could differentiate one from the other [6]. Hence, this algorithm is used in this study.

In this study, data augmentation techniques have been added as additional hidden layer of CNN structure, to explore the possibilities of producing better performance results of the ML model. Further performance indicators, such as mean squared error, precision and recall, and F1-score, have been used to support the finding.

The following sections discuss further about the number of related studies, followed by methodology and technology approach, then implementation, test result with discussion, and closed with conclusion and possible future work.

2. **Related Studies.** A number of studies have been done in digitizing and learning to classify batik motifs and patterns using a technological approach. Based on these studies, it can be seen how CNN architecture can be utilized to identify and predict the batik motifs and other image classifications scenarios. However, some limitations were still found from the studies as follows.

In 2017, a research study by Handhayani et al. [7] utilized shallow and deep learning approaches for classifying batik motifs. Another research study by Wicaksono et al. [8] also implemented a modified CNN architecture for batik motif image classification.

Based on these two studies, they successfully achieved an accuracy score for their classification model of up to 70% [7, 8]. It is believed that this low accuracy score was a result of the limitations of the training data set.

In addition, a number of research studies in computer vision have been implementated of CNN architecture, such as for land usage image classification scenario [6] and also even a teoritical research in modified CNN [9]. These studies have explored further the usage of CNN architecture to improve accuracy result for image classification scenario which can be adopted in our study.

As a continuation of previous work of [4, 10], in this paper, we are focusing on using Lasem-batik's motifs as the data set for a proposed ML model. Lasem-batik is known for its strong colors and complex motifs compared with other batik motifs in Central Java [11].

Following scenario study by [8] and modified CNN model [9], in this paper, we were implementing modified CNN architecture for image classification for our Lasem-batik motifs dataset with ADAM optimizer based on additional image augmentation and dropout layers. Image augmentation was implemented to increase the diversity of our batik motifs dataset for training models, without actually collecting new data.

Next, the following section will discuss more on the methodological and technological approaches.

3. **Methodology and Technology Approach.**

3.1. **Methodology.** As the methodology for this research, this research study implemented a quantitative method by conducting experimental testing for our ML model based on modified CNN architecture with and without image augmentation. Please refer to Figure 1 for the research methodology.

Based on Figure 1, we implemented a supervised learning by labeling the dataset based on five different classes based on the Lasem-batik motifs (Gunung Ringgit, Latohan, Kricak, Seritan and NyukPitu). Then, by utilizing Tensorflow 2.0 and Keras library, we used this labelled dataset for two different scenarios.

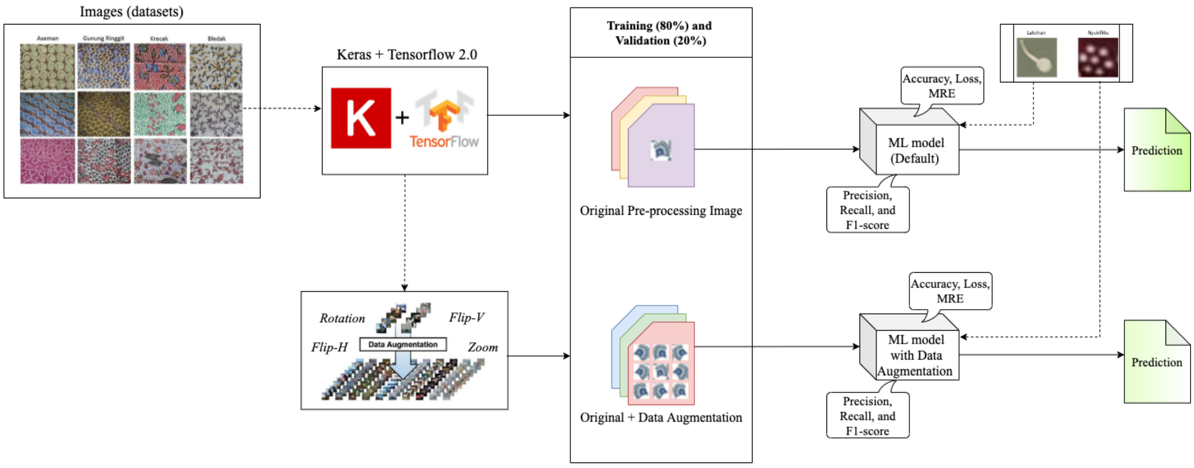

Figure 1. Research method of ML with modified CNN architecture with and without image augmentation

For the first scenario, we train and test the dataset with only using pre-processing dataset, while for the other scenario we added data augmentation layer into pre-processing dataset. Data augmentation layer has been added by calling augmentation functions such as “rotation”, “flip” and “zoom”.

After that, for each scenario, we divided 80% of the dataset for training and 20% of the dataset for validation. And we conducted a similar procedure for the model training steps and testing.

3.2. **Modified CNN based on image augmentation and dropout.** CNN or convolutional neural network is an image processing algorithm often used in computer vision. CNN is a multilayer system consists of the input layer, output layer, and a hidden layer that comprises multiple convolutional layers, pooling layers, fully connected layers [6].

In this study, all the CNN structure models were implemented by using Tensorflow version 2.0 and Keras library. These two libraries are based on Python and they are widely utilized to build training models to help data scientists to learn and predict [12].

This research study constructs two different CNN structure models, each for model without image augmentation, as shown in Figure 2 and the other with image augmentation, as shown in Figure 3.

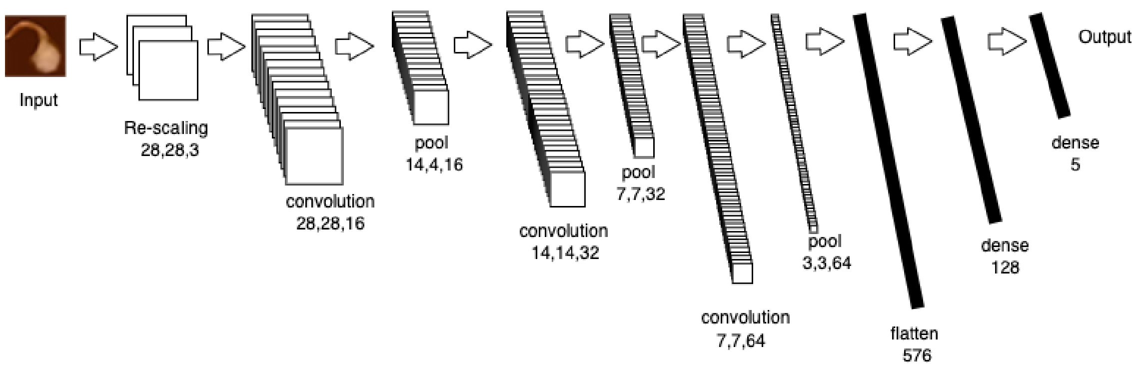

Figure 2. CNN structure model without image augmentation

In this structure model, it contains an input layer, a rescaling layer, six hidden layers composed of convolution and pool layers, a flatten layer and two dense layers before the output layer as the connection layer. The activation functions of the layers, except the output, are all rectified linear unit (ReLU) functions. The two-dimensional convolution

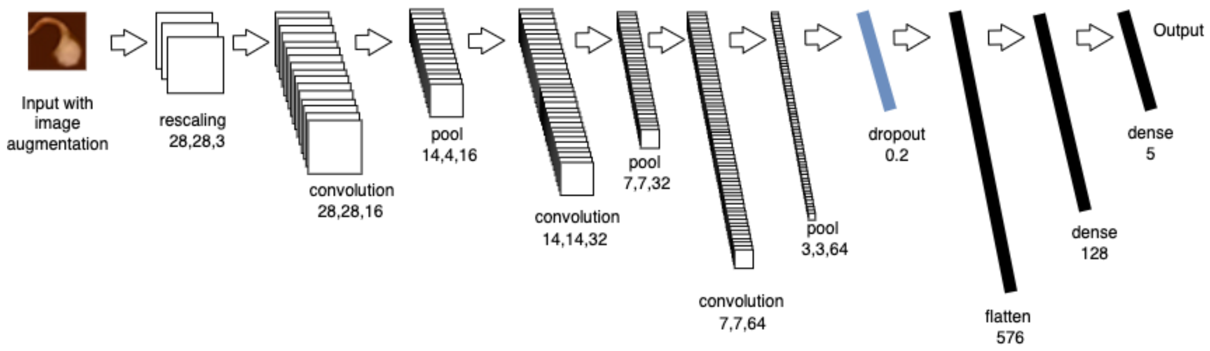

FIGURE 3. CNN structure model with image augmentation and dropout

operation is carried out by Conv2d function and pool operation by MaxPooling2D function from Tensorflow and Keras library.

The mathematical model representation utilized in this study follows a generalized computation node value function for neural network with multiple hidden layers and multiple nodes in each of the layers. The mathematical expression can be written as

$$a_n^L = \left[\sigma\left(\sum_m \theta_n m^L \left[\dots\left[\sigma\left(\sum_j \theta_{kj}^2 \left[\sigma\left(\sum_i \theta_{ji}^1 x_i + b_k^1\right)\right] + b_k^2\right)\right]\dots\right]_m + b_n^L\right)\right]_n$$

where we have a as units of hidden layer, b as units of next hidden layer, x as input layer, θ as weights matrix, and L layers with n nodes and $L-1$ layer with m nodes for a, and i, j, k nodes for b.

Hence, by following the same model representation mathematics, this second CNN structure is still implementing similar hidden layers from the first CNN structure model. However, we modified the CNN hidden layer by adding data augmentation layer as the additional input and also added Dropout with value before the connection layer.

The data augmentation layer works as image augmentation process that implements random vertical and horizontal flip functions, random zoom in/out functions and random rotation function. Further description on the process and the examples are discussed on the implementation section below.

3.3. **Evaluation techniques.** In order to evaluate the performance of our developed ML model, we measured the accuracy and loss scores for every iteration of training and validating the ML data set.

The mean squared error (MSE) of prediction was commonly used as a criterion for selecting variables [13]. This evaluation test contributes significantly in estimating the values of the predictor variables associated with future observations and the magnitude of the expected variance compared with commonly used criteria.

Next, the two evaluation methods "Precision" and "Recall", are commonly suggested as an appropriate evaluation of the classification of imbalanced data [14]. A perfect precision score of 1.0 means that every result retrieved by the model was relevant whereas a perfect recall score of 1.0 means that all relevant data sets were retrieved by the model.

Last method is "F1-score" where it can be used as an additional method that is commonly used to combine precision and recall into a single measure [15]. The next section describes further the machine learning model implementation and how it was tested.

4. **Implementation, Test Results and Discussion.** This section describes the implementation, test result, and discussion into three different parts based on the CNN structure model layers explained in Section 3.

4.1. **Part 1: Preparing and building an input pipeline.** The data sets that we used for this study were collected manually from a variety of Lasem-batik images. Around one hundred Lasem-batik linens were digitized and sliced into individual motif data with a specific size and resolution based on five different classes, which were labeled: 'Gunung Ringgit', 'Watu Pecah (Kricak)', 'Latohan', 'Nyuk Pitu', and 'Seritan' motifs. All images on the data sets were sliced into 28×28-milimeter squares and had a 72×72 resolution.

In addition, these data sets were also validated by a Lasem-batik expert to make sure of the validity and reliability. This step was important to ensure our ML model could train efficiently with high accuracy and low loss.

For setting up the environment, we started by creating an image classifier. For this purpose, we used a 'Sequential' model. Then, we imported our prepared data set from Lasem motif images that had been labelled. As mentioned previously, for the training, we used a total of 5671 Lasem motif images that had been grouped equally into 5 different sub-directories which were representing each label/class.

The pre-processing module from Keras library was used to build the data set as the input pipeline. For the initial steps, a number of parameters were set, such as batch-size and size of input image. In order to avoid over-fitting of the data, in this experiment, we set the batch-size value with a common mini-batch size of 32. We also set the dimensions for the input image value to 28×28 millimeter, which was following the minimum size of our data sets.

To avoid a bottleneck of data while we trained the data set in the model, there were configurations required to make sure of the stability of the training performance. The first configuration was utilizing buffered perfecting. This configuration was important in loading the data to avoid blocking when we yielded the data. For the second configuration, the cache was also implemented to make the images stay in memory after being loaded in the first place.

Normalizing the data sets was also an important step in preparing and building the input pipeline. Our data used the standard RGB (Red Green Blue) color format which has channel values from 0 to 255. This value was not ideal for a neural network. Hence, a rescaling layer was required to standardize the value into range from 0 to 1. For this purpose, a rescaling package from Keras could be utilized. In addition, since the data sets for this model had homogeneous sizes and dimension, we did not have to worry about resizing our data sets.

Before we continued the next stages, in investigating how well data augmentation worked for our model, we implemented and tested parts 2 and 3 below in parallel, by using the same pre-processing data sets as the input pipeline. Further detail about the implementation and test result for each stage is explained below.

4.2. **Part 2: Creating and compiling the model.** Based on the CNN structure model in Section 3, both our models were set to have three different two-dimensional convolution blocks followed by a max pool layer for each block. As a standard approach, three different filter values were set, which were 16, 32 and 64 units, to set up a different level of dimensional output for each convolution block. In addition, we also set up the 'relu' activation function for each block to fully connect with a 128-unit output.

After creating the model, the training and validation accuracy can be seen for each training iteration. For each iteration, it passed the metrics argument that had been compiled by optimizer 'Adam' and the losses 'SparseCategoricalCrossentropy' functions.

Then the model was run using training and validation data sets with the 'model.fit' function. In this experiment, the data was trained with 30 iterations as the default setting. However, we implemented the 'EarlyStopping' class from the Callbacks package. This class stopped the training when a monitor metric had stopped improving. This method was useful to avoid over-fitting in training iterations.

Using the ‘EarlyStopping’ class in the Callbacks, we could specify the variable that was being monitored. In this case, we used the ‘Validation-Loss’ variable to monitor the improvement metric. We set the mode with ‘min’ value to set the training to stop when the quantity monitored had stopped decreasing. We set the Patience properties to 3 as an accepted number of iterations with no improvement after which training would be stopped. Hence, we could get the number of actual iterations required for the model to train the data sets. Figure 4 shows the result of the training iterations.

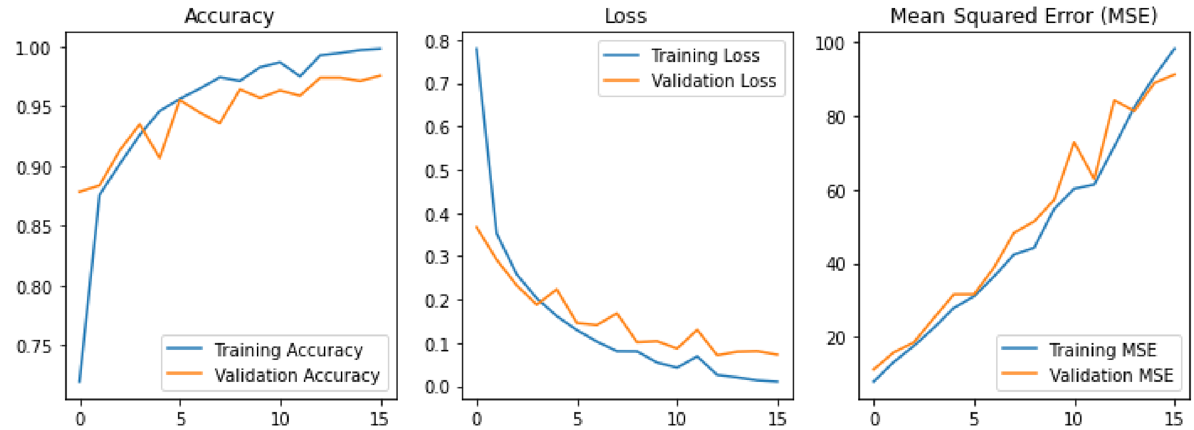

Figure 4. Accuracy, loss and MSE scores for default training data sets

Based on Figure 4, the training was stopped after 16 iterations. In this model approach, the accuracy score for training started from 71.9% and increased up to 99.7% in the 16th iteration. However, the accuracy score for validation was 87.8% in the beginning and only reached 97% in the 16th iteration. Based on loss scores, it showed an improvement from 77.8% to 1.03% for the training data sets and 36.7% to 7.2% for the validation data. Based on the results, it showed that both the training and validation accuracies were increasing linearly over time.

Based on the MSE scores between training and validation data sets, it can be seen that the training MSE had a slightly higher score compared with the validation data sets, except for the 15 iterations onward, it seemed the graph showed the opposite trend.

Even though there was no sign of the over-fitting phenomenon, the plots show that there were several spikes happened, especially for the validation curve. It showed that there was a drop of value in accuracy for a number of iterations.

We analyzed in more detail about the performance of the model without data augmentation. Further analysis was conducted to measure precision, recall and F1-score by using a classification report from the Sklearn metrics package. The results can be seen in Table 1 below.

Table 1. Performance evaluation of class name without data augmentation

	GnRinggit	Kricak	Latohan	NyukPitu	Seritan	Overall
Precision	0.944751	1	0.98227	1	0.952206	0.975845
Recall	1	0.984615	0.95189	0.985714	0.970037	0.978451
F1-score	0.971591	0.992248	0.966841	0.992806	0.961039	0.976905

This result was compared with the results from the model with data augmentation data sets explained in part 3 below.

4.3. **Part 3: Implementing data augmentation.** As mentioned earlier, this stage was conducted in parallel with part 2. In this stage, we explored the performance of the ML model with additional data augmentation in the training data sets.

Data augmentation is an optional technique that is normally used if any over-fitting phenomenon occurred because of the lack of data training examples. Basically, this approach is working by generating additional training data from the existing data set by creating random transformations (e.g., shifting, flipping, rotating, zooming, and changing the brightness level) that have trusted-looking images. This technique provides additional perspectives based on the same data sets.

The overall process in this stage was implementing four augmentation layers and adding dropout into the model pipeline, and then evaluating any possible improvement in the ML model performance.

The four different layers that were used in the model for data augmentation were 1) RandomFlip for Horizontal, 2) RandomFlip for Vertical, 3) RandomRotation, and 4) RandomZoom. In avoiding the loss of important information from the image for this experiment, we limited the random value for rotation and zoom to only 10% augmentation.

Next, in stage 3, we also implemented the same steps from stage 2 above. Hence, the result for the accuracy, loss and MSE result can be seen in Figure 5.

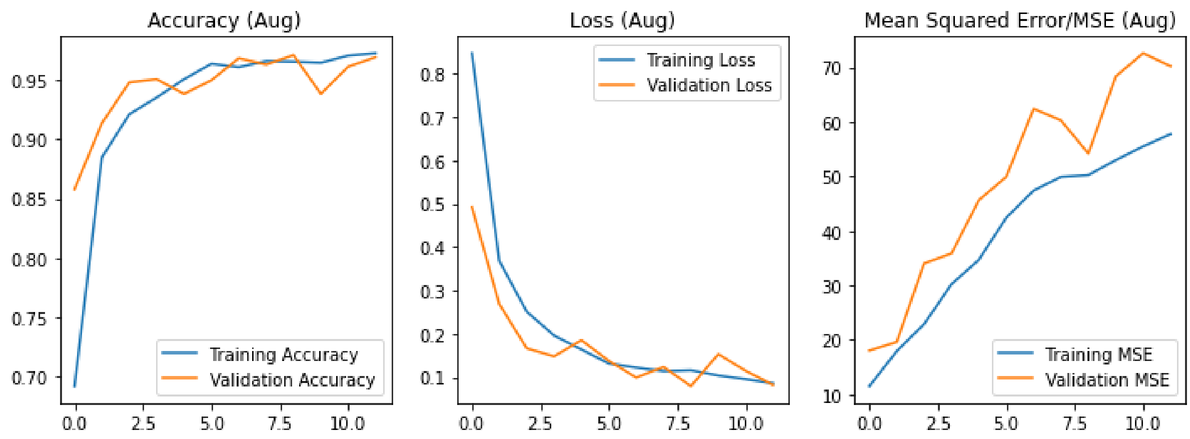

FIGURE 5. Accuracy, loss and MSE scores for training datasets with data augmentation

Based on Figure 5, the training for ML model with data augmentation was stopped after 11 iterations, five less compared with the first ML model. The accuracy score in this model started from 69.29% and increased to 97.2% for training and started from 85.8% to 96.9% for the validation accuracy test. There were fewer spikes and dropouts on the validation curve compared with the previous ML model. However, the loss scores only improved from 84.7% to 8.5% for the training data set and 49.5% to 8.1% for the validation data set.

The MSE scores between training and validation data sets showed higher scores compared with other models, and the gap between training and validation was also bigger compared with the other ML model.

Furthermore, similar analysis on the performance indicators on the model with additional data augmentation data sets can be seen in Table 2.

5. **Conclusion.** This paper is part of a research study that aimed to utilize advanced technology to digitize Indonesian batik motifs. The focus of the discussion in this paper was limited to the implementation of a machine learning (ML) model by using modified CNN model.

Table 2. Performance evaluation of class name with data augmentation

	GnRinggit	Kricak	Latohan	NyukPitu	Seritan	Overall
Precision	0.904762	0.994652	0.99635	1	0.945255	0.968204
Recall	1	0.953846	0.938144	1	0.970037	0.972406
F1-score	0.95	0.973822	0.966372	1	0.957486	0.969536

In this study, we have successfully adopted and implemented scenario study from [8] and modified CNN model [9] for our Lasem-batik motifs datasets. We have successfully investigated how well this technique could improve our ML model performance, focusing on two different scenarios, which are with and without data augmentation.

Based on the implementation, test results and discussion sections above, in this particular scenarios and data sets, the exploration of data augmentation techniques in training data sets showed no improvement in performance compared with the approach with data augmentation.

However, based on the training process, adding data augmentation into data sets could reduce the dropout and spikes in test results, especially for the validation variable. In addition, data augmentation also required fewer iterations before indicating the overfit trend in the training and validation process.

Based on the performance results of the overall ML model, for future studies and continuation, this developed ML model could be improved and tested further for an object detection model and classification of Batik Lasem motifs.

Acknowledgment. This research was funded by Directorate of Resources, Directorate General of Higher Education, Ministry of Education, Culture, Research, and Technology, Indonesia (Direktorat Sumber Daya Direktorat Jenderal Pendidikan Tinggi Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, Indonesia), with a research contract in the year 2021, No. 309E4/1AK/04/PT2021, date of 12 July 2021.

REFERENCES

[1] A. Malik, R. Retno and A. Ayu, The effect of wasp night composition on the making of Klowong batik on the quality of batik products, *Teknoin*, vol.22, no.6, 2016.

[2] A. Trixie, The philosophy of batik motifs as the identity of the Indonesian Nation, *Folio*, vol.1, no.7, pp.1-9, 2020.

[3] T. Handhayani, Batik Lasem images classification using voting feature intervals 5 and statistical features selection approach, *Proc. of 2016 International Seminar on Intelligent Technology and Its Application (ISITIA2016)*, DOI: 10.1109/ISITIA.2016.7828625, 2016.

[4] V. Basiroen, The acculturation of Batik Lasem's culture with Chinese motif in the 18th to 20th century, *Lekesan: Interdisciplinary Journal of Asia Pacific Arts*, vol.2, no.11, pp.59-67, 2019.

[5] I. Nurhaida, V. Ayumi, D. Fitrianah, R. Zen, H. Noprisson and W. Hong, Implementation of deep neural networks (DNN) with batch normalization for batik pattern recognition, *International Journal of Electrical & Computer Engineering*, vol.10, pp.2045-2053, 2020.

[6] S. Noppitak and O. Surinta, Ensemble convolutional neural network architectures for land use classification in economic crops aerial images, *ICIC Express Letters*, vol.15, no.6, pp.531-543, 2021.

[7] T. Handhayani, J. Hendryli and L. Hiryanto, Comparison of shallow and deep learning models for classification of Lasem batik patterns, *Proc. of 2017 1st International Conference on Informatics and Computational Sciences (ICICoS2017)*, no.10, pp.11-16, 2017.

[8] A. Wicaksono, N. Suciati, C. Fatichah, K. Uchimura and G. Koutaki, Modified convolutional neural network architecture for batik motif image classification, *IPTEK Journal of Science*, vol.2, no.8, DOI: 10.12962/j23378530.v2i2.a2846, 2017.

[9] Y. Jing and Y. F. Guanci, Modified convolutional neural network based on dropout and the stochastic gradient descent optimizer, *Algorithm Journal: Special Issues on Advanced Artificial Neural Network*, vol.11, no.3, pp.1-15, 2018.

[10] V. Basiroen, Creating Batik Lasem through a comparative study of Batik Lasem and Champa in the 15th to 19th century, *IOP Conference Series: Earth and Environmental Science*, vol.729, no.4, 2021.

[11] M. M. Sugarindra and E. A. Prakoso, Exploring the potential wealth of modern batik with the sido sublime motif Pakualaman batik association Yogyakarta, *Journal of Panrita Abdi*, vol.4, pp.245-254, 2020.

[12] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu and X. Zheng, TensorFlow: A system for large-scale machine learning, *Proc. of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16)*, pp.265-283, 2016.

[13] D. Allen, Mean square error of prediction as a criterion for selecting variables, *Technometrics*, vol.13, pp.469-475, 1971.

[14] B. Juba and H. Le, Precision-recall versus accuracy and the role of large data sets, *Proc. of the AAAI Conference on Artificial Intelligence*, vol.33, no.7, pp.4039-4048, 2019.

[15] S. Ding, Feature selection based F-score and ACO algorithm in support vector machine, *2009 2nd International Symposium on Knowledge Acquisition and Modeling (KAM2009)*, vol.1, pp.19-23, 2009.