Hyperparameter tuning of multilayer convolutional network and augmentation method for classification motive of batik

A. Nursikuwagus [a,1,*], T. Hartono [b,2], M. A. Nurwicaksono [b,3], M. M. Choir [b,4], M. A. Saputri [b,5]

[a] Department of Informatics Management, Universitas Komputer Indonesia, Bandung 40132, Indonesia
[b] Department of Information System, Universitas Komputer Indonesia, Bandung 40132, Indonesia
[1] agusnursikuwagus@email.unikom.ac.id ; [2] tono.hartono@email.unikom.ac.id; [3] agy@agyson.com;
[4] mugia.miftahul@gmail.com; [5] meyliaanggraenisaputri@gmail.com
* Corresponding Author

Received 26 October 2022; accepted 20 December 2020; published 10 January 2023

ABSTRACT

The purpose of this research is to create a batik motive image classification system to make it easier for the public to know the name of a type of batik motive. In carrying out this research, a quantitative method was used with seven kinds of batik motives that were augmented first, where 70% of the dataset was used for training and 30% for testing so that the accuracy and precision of the system were obtained. The result of this research is that the accuracy and precision of the system in classifying batik motive images is 0.985 or 98.5%. This high accuracy and precision were obtained because the quality of the previous dataset was improved by augmenting geometric and photometric. The machine learning method used was a Convolutional Neural Network which in previous studies also provided the highest accuracy and precision. The results of this study can be used for various purposes such as marketing, cultural reservation, and science.

KEYWORDS
Batik
Convolutional neural network
Image augmentation
Parameter Tuning
Comparison

This is an open-access article under the CC–BY-SA license

1. Introduction

Batik is a traditional Indonesian painting which is generally drawn on cloth media. Batik is an image of traditional patterns or patterns on a cloth painted using traditional methods. Etymologically in Javanese, Batik is defined as drawing dots on the fabric [1]. Historically, Batik in Indonesia is closely related to the development of the Majapahit Kingdom and its relation to the spread of Islam on the island of Java. In addition, in several writings, it is stated that the development of Batik was relatively rapid during the Mataram Sultanate. After that it continued during the Surakarta Sultanate and Yogyakarta Sultanate. On October 2, 2009, Batik was officially recognized by UNESCO as an Indonesian cultural heritage and a Humanitarian Heritage for Oral and Non-material Culture (Masterpieces of the Oral and Intangible Heritage of Humanity). And since then, October 2 has been celebrated as National Batik Day. In Indonesia, the number of various batik motives reaches thousands of types because each region has its batik motive. These motives can be divided into two main categories, geometric and non-geometric motives. Examples of geometric motives are Parang, Kawung, Megamendung, and Banji. Meanwhile, non-geometric motives include Betawi, Gurda, and Lunglungan. In Batik, this motive is a characteristic that identifies the type of Batik. These patterns are usually arranged repeatedly to describe the primary mark on the fabric.

The diversity of batik motives in Indonesia can potentially increase the creative industry and tourism sectors. Many domestic and foreign tourists are interested in Batik and want to know more about batik information such as its history, methods, places of manufacture, etc. However, the difficulty of distinguishing types of Batik based on very diverse motives is still a problem. Therefore, we need a system to help classify Indonesian batik motives. Data classification is intended to identify the characteristics of objects in the database and categorize them into different groups. The purpose of classifying Batik is to divide the image of Batik into motive classes according to the pattern of motives so that they are easier to identify according to their characteristics [2].

Data augmentation is the process of changing data or usually in the form of existing images, into new data without changing the essence of the data [3]. Data augmentation is carried out to avoid overfitting by enriching the dataset by making various other versions of the original image in the training dataset to improve the quality of learning in the system. Image augmentation can be divided into two categories, Geometric Transformation and Photometric Transformation. Geometric Transformation is usually used to deal with positional bias. For example, in face detection programs where each face in each image is in a different position, geometric transformation can help the program determine the exact face location [4]. Meanwhile, photometric adaptation is usually used to handle differences in color or image quality so that CNN can study different versions of the color in the image. The main problems that occur due to the application of image augmentation are the increased storage requirements, transformation load, and training time [5].

In this study, we propose classifying Batik using Convolutional Neural Network CNN Convolutional Neural Network (CNN) is the result of the development of the Artificial Neural Network (ANN), so CNN is also a processing system that works and is inspired by the workings of the nervous system of living things, for example, in the human brain. The central concept of how the Neural Network works itself is to enter input which is usually an image or multidimensional vector, and this is done by the input layer, which will pass it on to hidden layers that are interconnected to perform a series of decisions so that each layer can learn from the previous layer until it finally produces output in the form of classification information. CNN was developed to find a more efficient method than traditional ANN. The solution applied is that neurons on CNN will only connect to specific areas in the previous layer so that the connection from layer to layer will be more efficient than the ANN method, which will make a complete connection. This simple difference reduces the number of parameters needed to set up the model, and the architecture formed is also simpler [6]. In addition to the input and output sections, the essential components of CNN consist of three types of layers, namely the convolutional layer, pooling layer, and fully connected layer [4]. To work optimally, CNN requires many training datasets to avoid overfitting (a situation where the NN cannot learn effectively, thereby reducing the precision of qualification or prediction) [1]. This problem is usually solved by augmenting the existing dataset.

Several previous studies have also attempted to classify batik images, such as that conducted by A Haris Rangkuti using the Fuzzy Neural Network, which provides a precision of 94% [7]. Then a slightly higher accuracy was obtained by Taufiqul Bariyah using the Convolutional Neural Network, where in the preprocessing, image augmentation in the form of cropping and resizing, resulting in a precision of 94.5% [8]. In this study, we want to follow up on previous research by adding more image augmentation, both geometrically and photometric transformation, and reverting to using a convolutional neural network which previously produced the highest precision.

In our study, the data used is more geometric, which helps the program determine the pattern's position as well as using photometric that are commonly used to handle differences in color or image quality. This study achieved exquisite precision, reaching 98.5%. Compared to previous studies, the study outperforms than previous especially on improving the quality of the dataset to get excellent precision.

2. Method

This research was conducted through a quantitative approach, where datasets were taken from various sources on the Internet. The methods used in this study include the acquisition of batik datasets from various sources (Scraping from the Google Search Engine and Retrieval of datasets from the Kaggle dataset repository), sorting of batik datasets, up to the

Jurnal Informatika
Vol. 17., No. 1, January 2023, pp. 28-38

batik classification process according to the motive using Neural Network and Testing models as shown in Fig. 1.

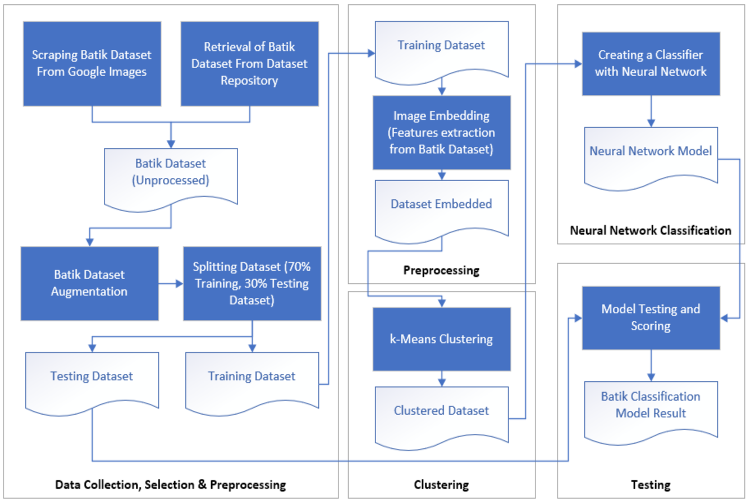


Fig. 1. Research Flow, separated in three blocks are Data Collection, Selection and Processing block, Clustering block, Testing block

2.1. Batik Dataset Collection

In the first stage, the batik datasets were searched and collected through two sources, namely, image scraping from the google search engine with the keyword "Batik Motive" and batik dataset retrieval from the Kaggle repository. The initial batik dataset collected from various sources amounted to 215 batik images. Dataset image retrieval is based on 7 batik motives: Banji, Betawi, Ceplok, Gurda, Kawung, Megamendung and Parang, as shown in Fig. 2.

Fig. 2. Example Dataset of Batik Motive

2.2. Image Augmentation

At this section, we explained the flow of process image augmentation. Beginning from image reduction to obtain uniformly image size, image pre-processing, distribution of dataset between dataset training and dataset testing, image embedding or feature extraction, clustering, convolutional neural network, model metric for the architectures.

2.3. Image Pre-processing

A collection of batik datasets with different image sizes needs to be resized to make them uniform. Uniformity of image size is carried out in the second stage, namely resizing/resizing the image to 256 x 256px (pixels/pixels). The image size of 256 pixels was chosen because it is the optimal size for conducting training datasets on the Convolutional Neural Network

ISSN 2086-8138 (print) | 2745-7249 (online)

(CNN) based on research conducted by Daisuke Hirahara et al. (2020) regarding the effect of the amount of data and image scaling on deep learning training [9] and research conducted by JGA Barbedo on the impact of dataset size and variation on the effectiveness of deep learning [10]. Image size changes are performed using a script with the Python programming language, as shown in Fig. 3.

```python
import os
import sys
from PIL import Image

path = "E:\\PROJECT\\DATA MINING\\clean-batik-dataset\\batik-megamendung\\"
dirs = os.listdir(path)

def resize():
    for item in dirs:
        if os.path.isfile(path+item):
            im = Image.open(path+item)
            f, e = os.path.splitext(path+item)
            imResize = im.resize((256,256),  Image.LANCZOS)
            imResize.save(f + ' resized.jpg', 'JPEG', quality=90)

resize()
```

Fig. 3. Script code for resizing the image

The next step is to increase the data variance using augmentation algorithms such as cropping, gaussian blur, hue, inverting color, saturation, inverting and rotating images. An example of this augmentation is shown in Fig. 4.

Fig. 4. Example of Image Augmentation of Megamendung Batik

By increasing the batik dataset's variants, the CNN image input will be more diverse. Thus, the training results carried out by the neural network are expected to be more accurate because there are other alternative images in one image of a batik motive. The number of images from the dataset after the augmentation process has reached 4,047 images. This 4,047-image dataset was generated from 19 augmentation algorithms applied to 215 initial batik motive datasets.

2.4. Dataset Distribution

At this stage, the distribution of datasets for training and model testing purposes is carried out in a proportion of 70-30 (70% for training datasets, 30% for testing datasets). This dataset's distribution is carried out to produce an accurate and independent CNN model for the later model testing stage. The proportion of 70-30 used in this study is based on research which shows that the best results are obtained when using 20-30% of the data for testing, and the remaining 70-80% of the data is used for training [11], [12] after the sequential distribution process is 1,214 testing datasets and 2,833 training datasets.

 Jurnal Informatika
Vol. 17., No. 1, January 2023, pp. 28-38

2.5. Image Embedding

The next step is to perform Image Embedding or feature extraction from batik datasets that have been augmented. This step is carried out to calculate the feature vector for each image from the batik dataset and produce an enhanced data table with an image descriptor column. Image Embedding was carried out using the VGG16 deep learning model proposed by Karen Simonyan and Andrew Zisserman from the Visual Geometry Group Lab, Oxford University, in 2014 in their paper entitled "Very deep convolutional networks for large-scale image recognition" [13]. VGG-16 is a Convolutional Neural Network (CNN) which consists of 16 layers. VGG-16 was trained using the ImageNet dataset, a collection of more than 14 million images belonging to 22,000 categories [14]. The VGG-16 architecture is illustrated as shown in Fig 5. The filters used in VGG16 are convolutional/filter/weights measuring 3x3 and 1x1 with stride one, and Max-Pooling measuring 2x2 with stride 2. The selection of pooling in the pooling layer section is to use Maxpooling2D. This Max-Pooling technique is to down sample by utilizing the maximum value when doing windowing and stride. Maxpooling2D uses a large 2x2 matrix. The output of this down sampling is a matrix measuring 112x112 with 64 channels.

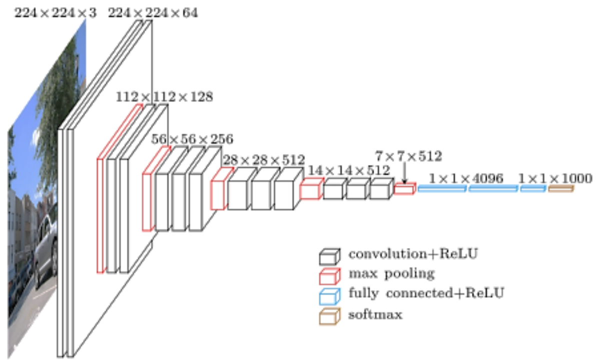


Fig. 5. VGG16 Architectures

2.6. K-Means Clustering

In 1984 K-Means was first introduced by Stuart Lloyd. K-Means is a grouping algorithm that is widely used. K-Means works by grouping existing objects into groups, commonly called segments, so objects in each group are more similar than objects in a different group. K-means uses a Euclidean distance measure and iteratively determines each record in the source set. The procedure begins by selecting k with the initial record as the cluster's centre (initial seed) and determining each record closest to the cluster. A new record is added to the cluster, and its centre is recalculated to reflect its new members. This iterative process is repeated until it converges and the log migration with the cluster no longer loses its solution. K-means is a heuristic algorithm that divides the data set into K groups by minimizing the sum of the squares of the distances in each group. The basic principle of this technique is to construct K partitions/centroid/mean (mean) of the data set. The K-Means algorithm starts by forming a cluster partition and then iteratively refines the partition until there is no significant change in the partition [15].

This algorithm for K-Means algorithm run is as follows:

- Specify number of K (clusters).
- Firstly, initialize the center, we can do by shuffling the dataset and select randomly K data points without replacement.
- Iterative until there is no change the centroids point.
  - Compute the sum of the squared distance between data points and all centroids.
  - Assign each data point to the closest cluster (centroid).

- Compute the centroids for the clusters by taking the average of the all-data points that belong to each cluster.

Data clustering using the K-Means Clustering method is generally carried out with a basic algorithm, starting from determining the number of clusters and allocating data into clusters randomly. Calculate the centroid/average of the data in each cluster, allocate each data to the nearest centroid/average, and then return to Step 3. If there is still data moving clusters or if the centroid value changes, there is an above the specified threshold value or if the change in the value of the objective function used is above the specified threshold value.

K-means clustering is used at this stage to cluster batik types based on their characteristics. K-means Clustering is a data analysis method or Data Mining method that performs the modelling process without supervision (unsupervised) and is one method that performs data grouping with a partition system [16]. The K-means algorithm is a non-hierarchical method that uses most population components as the initial cluster center. The cluster center is randomly selected from the population data set in this phase. Next, K-Means examines each component in the data population and assigns the component to one of the specified cluster centers based on the minimum distance between the components and all clusters. The position of the cluster center is recalculated until all data components are in each cluster center, and finally, a new cluster center position is formed [17].

2.7. Convolutional neural network classifier

The next step is classifying batik using the Convolutional Neural Network (CNN). This stage will produce a model that can be used to predict the type of batik based on input in batik motives. It consists of 3 types of layers (layers) – input layer (input layer), output layer (output layer) and hidden layer (hidden layer) [18], as shown in Fig. 6.

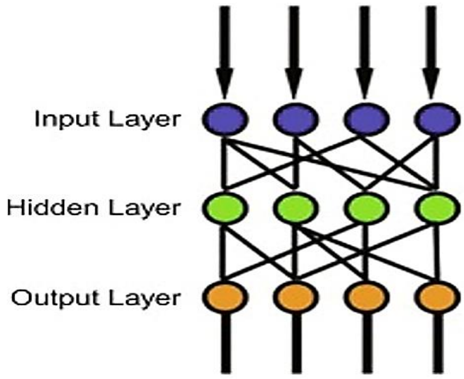


Fig. 6. Convolutional Neural Network layers

In this stage, the CNN model architecture contains 100 hidden layers, using ReLu as an activator and SGD as a solver and optimizer. The accuracy obtained from using a neural network depends on the number of layers but also the type of activation function used. The activation function is useful so that the CNN network is more dynamic and can get more complex information [15], as in our case, namely image classification, by introducing a non-linear property on the network [19]. In simple language, the activation function is useful for converting input into output which will then be input back into the next layer in the hidden neural network stack. If the activation function is not used, the output of each layer will only be a linear function, which means that it is only a polynomial of the power of 1 [15].

For the sake of the performance and quality of the CNN results, selecting the type of activation function used is very influential. This selection process takes a long time by trying different activation functions and comparing the results or researching similar cases. However, in most cases, the ReLU (Rectified Linear Unit) method gives better results than other methods, so we can try using the ReLU method first, and if the results are not satisfactory, we can try other methods [15]. And in our case, the use of ReLU has also proven to give the most satisfactory results, so the activation function we use in this model is ReLU.

ReLU is the most widely used activation function according to [15] , so since 2012, ReLU has become the default activation function in the deep learning community [19]. ReLU is a non-linear activation function that has the advantage of its performance efficiency. This is because not all neurons in the network are activated simultaneously [15]. The way ReLU works is to maintain the positive value of and change the negative value to zero [20], so mathematically, ReLU can be defined as follows:

$$f(x) = Max\ (0\ , x) \tag{1}$$

One of the problems of using a neural network is a large burden to run backpropagation on the training dataset so that the process is not optimal [21]. The Optimization function is a solution to the problem by applying an algorithm or method that ultimately determines how fast the learning process of a system is, and this speed is commonly known as the learning rate. So that the selection of the type of optimization function used will have a major effect on system performance [20]. In this case, we apply an optimization function of the Stochastic Gradient Descent (SGD) type, which has a learning rate of 0.1 [19]or 0.01 [20], which by default does not apply momentum. The learning rate can be adjusted to each case. In this study, a learning rate of 0.01 was used. SGD is a variation of another type of optimization function, gradient descent, with the concept of updating parameters for each training data. Still, SGD does not loop, so it is faster than the original version [20]. This process can be defined using the equation:

$$\theta = \theta - \eta * \nabla\theta J\ [\theta; x(i); y(i)] \tag{2}$$

where θ is parameter of the update result is the amount of learning rate used, $x(i)$ and $y(i)$ this data being trained.

SGD works at each optimization step by using the partial derivative estimator, substituting the exact partial derivative, and when the estimator is unbiased, making it possible to prove strict convergence guarantees in a simplified setting. SGD is also a common method of choice for most large-scale machine-learning models because it has better speed in gradient evaluation and convergence [22].

2.8. Metric

The final stage is testing (testing) and scoring (assessment) the accuracy level of the model built using CNN. The model generated from the CNN process is tested by testing the dataset that has been prepared previously. The results of this test will be summarized in the form of a Confusion Matrix according to the format shown in Fig. 7.

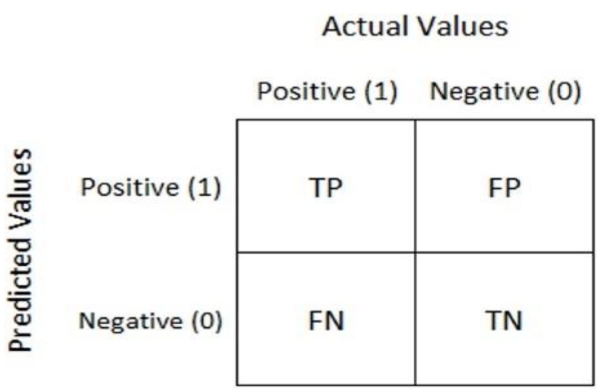


Fig. 7. Confusion matrix

Confusion Matrix is a performance measurement for classification using machine learning, where the output is a table with 4 different combinations of predicted and actual

Vol. 17., No. 1, January 2023, pp. 28-38
ISSN 2086-8138 (print) | 2745-7249 (online)

values. Based on the Confusion Matrix, the level of accuracy (3), precision (4), recall (5) and F1-Score (6) is calculated using the following formula:

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

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

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

$$F1 - Score = \frac{2}{\frac{1}{Recall}+\frac{1}{Precision}} \quad (6)$$

Where TP, TN, FN, and FN stand for True Positive, True Negative, False Negative, and False Positive, respectively.

3. Results And Discussion

The batik image classification model was tested using the Random Sampling method with 10 times Repeat train/test parameters, 70% training set size and stratified sampling. The results of model testing are displayed in tabular form as a Confusion Matrix, as shown in Fig. 8. As shown in Fig. 8, the model was tested 12,150 times and divided into seven types of batik. The purple cell-matrix indicates that the predictions made on this type of batik follow the training dataset or can be said to be actual.

Actual \ Predicted	batik-banji	batik-betawi	batik-ceplok	batik-gurda	batik-kawung	batik-megamendung	batik-parang	Σ
batik-banji	1132	1	5	0	0	2	0	1140
batik-betawi	5	1697	3	1	1	1	2	1710
batik-ceplok	7	6	1412	12	21	0	22	1480
batik-gurda	2	9	5	1178	0	2	4	1200
batik-kawung	2	3	13	1	1977	4	0	2000
batik-megamendung	0	5	8	5	2	2320	0	2340
batik-parang	0	8	2	5	6	8	2251	2280
Σ	1148	1729	1448	1202	2007	2337	2279	12150

Fig. 8. Confusion Matrix for each classification. The study defined seven class as the matrix shows. The purple color stands for the true predictive class and the other is the error class. Five metrics such as accuracy under curve (AUC), accuracy (acc), F1-measures (F1), precision (Prec), and Recall (Rec)

Another cell-matrix indicates the inaccuracy of the predictions made. The average level of accuracy, precision, recall and F1-Score also has very good results, which is 98.5% accurate. The accuracy, precision, recall and F1-Score values for each type of batik are shown in Table 1.

Table 1. Model metrics

Batik	AUC	Acc	F1	Prec	Rec
Batik Banji	0.999	0.998	0.990	0.986	0.993
Batik Betawi	1.000	0.996	0.987	0.981	0.992
Batik Ceplok	0.994	0.991	0.964	0.975	0.954
Batik Gurda	1.000	0.996	0.981	0.980	0.982
Batik Kawung	0.998	0.996	0.987	0.985	0.989
Batik Megamendung	1.000	0.997	0.992	0.993	0.991
Batik Parang	0.999	0.996	0.987	0.988	0.987
Average	0.999	0.985	0.985	0.985	0.985

Jurnal Informatika
Vol. 17., No. 1, January 2023, pp. 28-38

However, there are several datasets of batik motives that the model cannot predict correctly. This can happen because the tested batik motive dataset contains several types of batik in one image. As seen in Fig. 9, where the batik motives displayed contain 2 types of batik, namely Ceplok Batik and Parang Batik.

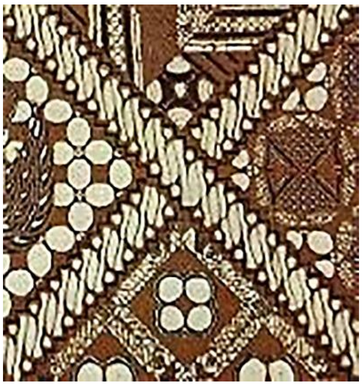
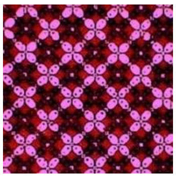


Fig. 9. The motive contains two types of batik: Ceplok and Parang

Based on the test results, the augmentation of the batik image that was carried out was confusing for the model testing the sample dataset. The following are some sample datasets whose test results are confusing for the model shown in Table 2.

Table 2. Comparison results between prediction and actual. The result shows there is still error in prediction the image using the model

Motive of Batik	Prediction	Actual
	Batik Ceplok	Batik Banji
	Batik Betawi	Batik Gurda
	Batik Ceplok	Batik Kawung
	Batik Kawung	Batik Ceplok

It can be seen from the second example in Table 2 the model predicts that the image of the batik motive being tested is Batik Betawi. The actual batik motive is Batik Gurda. The algorithm multiply used in image augmentation changes the color of the Batik Betawi dataset to resemble the Batik Gurda dataset, as shown in Fig. 10 [4].

(a) (b)

Fig. 10. The image augmentation process makes Batik Betawi, and Batik Gurda look similar , (a) Betawi Batik with various augmentation, and (b) Batik Gurda with various augmentation

4. Conclusion

From the results of this study, the Convolutional Neural Network can be applied to the classification problem of batik motives and shows a very good performance with an average accuracy rate of 98.5%, AUC of 99.9%, precision level of 98.5%, recall of 98.5%, and F1-score of 98.5%. This study uses 7 types of batik motives: Banji, Betawi, Ceplok, Gurda, Kawung, Megamendung and Parang. The initial dataset containing batik motives amounted to 215 images, and after preprocessing with 19 augmentation algorithms, it became 4,047 images of batik motives. Then, the augmented dataset is divided into 2 types of datasets, namely testing and training testing datasets, which are 1,214 and 2,833, respectively.

However, despite the high accuracy obtained, there are still errors or misclassifications in predicting batik motives from the testing dataset. This is due to the augmentation algorithm, which, when applied to an image of a particular motive, will resemble the image of other dissimilar batik motives. Then, the next cause of misclassification is because the image dataset of certain batik motives is a mixture of 2 or more types of batik. This will make the model doubt in determining the type of batik that should be chosen. This research is not perfect, so it needs to be improved for the effectiveness of the augmentation algorithm used in the future. Thus, errors or misclassification of the model can be minimized.

Acknowledgment

Our special thanks to Prof. Dr. H. Eddy Soeryanto Soegoto as a Rector Universitas Komputer Indonesia who supports and funds our research and disseminates the international seminar. Our special thanks to all data mining subject course students who help us to collect the dataset and annotate the image. For Dean of Faculty Engineering and Computer Science Dr. H. Herman Soegoto give the privileges to use the material computing and field. Dr. Wartika chief of informatics management and information system who gave permitted to make research on surrounding batik as an Indonesian culture.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. None of the authors have received any funding or grants from any institution or funding body for the research.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] Y. M. K. Syabana and G. B. Park, “A study on the applicability of batik for public transportation design in Indonesia,” *J. Mechatronics, Electr. Power, Veh. Technol.*, vol. 11, no. 2, pp. 75–85, Dec. 2020, doi: 10.14203/j.mev.2020.v11.75-85.

[2] K. Nuringsih, C. Cokki, M. N. Nuryasman, and H. Mularsih, “Behind the Pattern: Maintaining the Sustainability of Local Cultural Wisdom in Batik Entrepreneurial Sector,” in *Proceedings of the Ninth International Conference on Entrepreneurship and Business Management (ICEBM 2020)*, May 2021, vol. 174, pp. 1–9, doi: 10.2991/aebmr.k.210507.001.

[3] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, Aug. 2017, vol. 2018-Janua, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.

[4] J. Gu *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018, doi: 10.1016/j.patcog.2017.10.013.

[5] A. Mikolajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, May 2018, pp. 117–122, doi: 10.1109/IIPHDW.2018.8388338.

[6] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[7] A. H. Rangkuti, “Classification of Batik Motifs Based on Character Resemblance to Wavelet Transform and Fuzzy Neural Networks,” *ComTech Comput. Math. Eng. Appl.*, vol. 5, no. 1, p. 361, Jun. 2014, doi: 10.21512/comtech.v5i1.2630.

[8] T. Bariyah, M. A. Rasyidi, and N. Ngatini, “Convolutional Neural Network for Multi-Label Classification Method on Batik Motifs,” *Techno.Com*, vol. 20, no. 1, pp. 155–165, Feb. 2021, doi: 10.33633/tc.v20i1.4224.

[9] D. Hirahara, E. Takaya, T. Takahara, and T. Ueda, “Effects of data count and image scaling on Deep Learning training,” *PeerJ Comput. Sci.*, vol. 6, p. e312, Nov. 2020, doi: 10.7717/peerj-cs.312.

[10] J. G. A. Barbedo, “Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification,” *Comput. Electron. Agric.*, vol. 153, pp. 46–53, Oct. 2018, doi: 10.1016/j.compag.2018.08.013.

[11] Y. Xu and R. Goodacre, “On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning,” *J. Anal. Test.*, vol. 2, no. 3, pp. 249–262, Jul. 2018, doi: 10.1007/s41664-018-0068-2.

[12] B. Vrigazova, “The Proportion for Splitting Data into Training and Test Set for the Bootstrap in Classification Problems,” *Bus. Syst. Res. J.*, vol. 12, no. 1, pp. 228–242, May 2021, doi: 10.2478/bsrj-2021-0015.

[13] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Sep. 2014, pp. 1–14. [Online]. Available at: https://arxiv.org/abs/1409.1556v6.

[14] S. Tammina, “Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images,” *Int. J. Sci. Res. Publ.*, vol. 9, no. 10, p. p9420, Oct. 2019, doi: 10.29322/IJSRP.9.10.2019.p9420.

[15] M. Ahmed, R. Seraj, and S. M. S. Islam, “The k-means Algorithm: A Comprehensive Survey and Performance Evaluation,” *Electronics*, vol. 9, no. 8, p. 1295, Aug. 2020, doi: 10.3390/electronics9081295.

[16] E. L. Lydia, S. Kannan, S. SumanRajest, and S. Satyanarayana, “Correlative study and analysis for hidden patterns in text analytics unstructured data using supervised and unsupervised learning techniques,” *Int. J. Cloud Comput.*, vol. 9, no. 2/3, p. 150, 2020, doi: 10.1504/IJCC.2020.109373.

[17] K. Peng, V. C. M. Leung, and Q. Huang, “Clustering Approach Based on Mini Batch Kmeans for Intrusion Detection System Over Big Data,” *IEEE Access*, vol. 6, pp. 11897–11906, Feb. 2018, doi: 10.1109/ACCESS.2018.2810267.

[18] A. Salma, “Implementation of Multilayer Perceptron for Image Classification,” *Proceeding Int. Conf. Sci. Eng.*, vol. 4, pp. 212–215, Feb. 2021, Accessed: Aug. 01, 2021. [Online]. Available: https://sunankalijaga.org/prosiding/index.php/icse/article/view/660.

[19] H. H. Chieng, N. Wahid, O. Pauline, and S. R. K. Perla, “Flatten-T Swish: a thresholded ReLU-Swish-like activation function for deep learning,” *Int. J. Adv. Intell. Informatics*, vol. 4, no. 2, p. 76, Jul. 2018, doi: 10.26555/ijain.v4i2.249.

[20] N. D. Miranda, L. Novamizanti, and S. Rizal, “Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50,” *J. Tek. Inform.*, vol. 1, no. 2, pp. 61–68, Dec. 2020, doi: 10.20884/1.jutif.2020.1.2.18.

[21] J. Yang and G. Yang, “Modified Convolutional Neural Network Based on Dropout and the Stochastic Gradient Descent Optimizer,” *Algorithms*, vol. 11, no. 3, p. 28, Mar. 2018, doi: 10.3390/a11030028.

[22] R. Sweke *et al.*, “Stochastic gradient descent for hybrid quantum-classical optimization,” *Quantum*, vol. 4, p. 314, Aug. 2020, doi: 10.22331/q-2020-08-31-314.