IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 5, October 2025, pp. 3970~3981
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i5.pp3970-3981

Optimizing nitik batik classification through comparative analysis of image augmentation

Suprapto[1], Meilany Nonsi Tentua[2], Ahmad Rizki Maulana[1]
[1]Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia
[2]Department of Informatics, Faculty of Science and Technology, Universitas PGRI Yogyakarta, Yogyakarta, Indonesia

Article Info

Article history:

Received Nov 21, 2024
Revised Jul 4, 2025
Accepted Aug 6, 2025

Keywords:

DCGAN
Geometric transformation
Image data augmentation
Intensity transformation
Nitik batik motif

ABSTRACT

Nitik batik is one of the most intricate and culturally significant motifs in Yogyakarta's batik tradition, characterized by its complex, geometric dot-based patterns. The unique challenges of automatically classifying nitik batik motifs stem from the high variability within the class and the limited availability of training data. This study investigates how different image data augmentation techniques can enhance the performance of a random forest classifier for nitik batik motifs. Techniques such as geometric transformations (flip, rotate, and scaling), intensity transformations (cut-out, grid mask, and random erasing), non-instance level augmentation (pairing samples), and unconditional image generation (deep convolutional generative adversarial network (DCGAN)) were used to expand the dataset and improve the model's ability to generalize. The results show that specific techniques, notably flip, cut-out, and DCGAN, significantly improved classification accuracy, with flip achieving the highest accuracy improvement of 20.20%, followed by cut-out at 19.27% and DCGAN at 16.25%. Moreover, DCGAN demonstrated the lowest standard deviation (0.78%), indicating high stability and robustness in classification performance across multiple validation folds. These findings suggest that augmentation techniques effectively improve classification accuracy and enhance the model's ability to generalize from limited and complex datasets.

Corresponding Author:

Suprapto
Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada
North Sekip, Bulak Sumur, Yogyakarta 55281, Indonesia
Email: sprapto@ugm.ac.id

1. INTRODUCTION

Batik is Indonesia's rich and diverse cultural heritage, with motifs that reflect regional identities and high aesthetic values [1]. Automatic classification of batik motifs is vital in cultural preservation, collection management, and developing image-based applications related to the creative industry [2]. One of the oldest batik motifs typical of Yogyakarta is nitik batik. Nitik batik motifs are complex motifs consisting of thousands of dots arranged and measured in such a way as to form geometric spaces, angles, and fields [3]. The classification of nitik batik motifs faces challenges, including high intraclass variation, pattern complexity, and limited data available for model training [4].

In computer vision, image classification techniques have overgrown owing to advances in machine learning algorithms and the availability of extensive data. Some batik classification modeling has been done using machine learning algorithms such as k-nearest neighbors (KNN) [5], support vector machines (SVM) [6], backpropagation neural networks (BNN) [7], and decision trees [8]. Meanwhile, classification using deep learning algorithms for batik modeling mainly uses convolutional neural networks (CNN) [9]. The performance of classification models is highly dependent on the quality and quantity of data used during the training process [10], including batik classification modeling. Image data augmentation is an effective method to improve model performance [11]. Image data augmentation manipulates the original data to produce new variations that can enrich the dataset. The model can learn better and generalize more to data that has never been seen before [12].

Augmentation techniques can be categorized into several groups, including geometric transformation, intensity transformation, non-instance level augmentation, and unconditioned image generation [13]. Geometric transformation [14] and intensity transformation [15] are often used in augmentation techniques in batik classification modeling. Both augmentation techniques can increase the invariance of the model to changes in position. They can help the model deal with lighting and texture variations in the original image. However, unconditional image generation augmentation techniques have yet to be used for data addition in classification modeling, especially nitik batik. Unconditional image generation techniques, such as deep convolutional generative adversarial network (DCGAN), only reproduce new images to produce previously unknown batik patterns [16], [17].

This study aims to fill this gap by conducting a comparative study of the augmentation methods and analyzing their impact on the accuracy and robustness of the nitik batik motif classification model. The use of public datasets on nitik batik cloth motifs provides an opportunity to comprehensively evaluate how these augmentation techniques affect the performance of the classification model. The goal is to understand better how these techniques can enhance the effectiveness and efficiency of the nitik batik classification system. Our contribution is summarized as follows:

– The study introduces and compares the effectiveness of multiple image augmentation techniques—geometric transformations, intensity transformations, non-instance level augmentation, and unconditioned image generation—for improving nitik batik classification. By exploring a wide range of augmentation methods, the study provides a comprehensive understanding of how different techniques impact model accuracy and stability.
– By applying the random forest classifier in combination with augmentation techniques, the study demonstrates significant improvements in classification accuracy, especially through techniques like flip, cut-out, and DCGAN.
– The study emphasizes not only the accuracy but also the stability of the model, as measured by the standard deviation of cross-validation accuracy. The findings show that certain augmentation techniques, particularly DCGAN, provide high stability, which is crucial for deploying reliable classification models in practice.

2. RELATED PAPER

Data augmentation is essential in overcoming dataset limitations by artificially expanding the training data, while various modeling techniques have emerged to improve image classification performance. This review examines recent research's role in data augmentation and batik modeling strategies. Across the studies reviewed, data augmentation geometric transformations techniques were frequently used for classification model of batik, such as flipping [14], [18], [19], rotation [20], scaling [21], [22], shearing [23], and noise injection [24] to improve model generalization and reduce overfitting. More advanced methods, such as random erasing data [25] and brightness modulation [15], [26], are implemented to input data variations.

Various classification methods have been applied to automate the recognition of batik motifs, each addressing the unique challenges posed by the complex and highly detailed patterns in batik fabrics. Traditional machine learning techniques, such as SVM [27], KNN [28], BNN [29], and decision tree [8] have been widely used for batik classification due to their simplicity and effectiveness in handling small-scale datasets. More recently, deep learning approaches, particularly CNN, have gained prominence in batik classification tasks [2], [9], [30]. CNNs automate the feature extraction, learning intricate patterns directly from raw images, making them highly effective for complex batik motifs, including nitik batik [31]. The ability of CNNs to capture multi-level features, from edges to textures, has significantly improved the accuracy and efficiency of batik motif classification. However, CNNs often require large amounts of labeled data, which can be a limitation for batik datasets.

In addition to CNNs, random forest have been employed for Surakarta batik fabric classification due to its robustness and ability to handle small, imbalanced datasets [32]. Random forest creates an ensemble of decision trees, each trained on different parts of the data, allowing the model to capture various features from batik motifs and providing stable predictions even with limited data. The existing literatures show that image data augmentation significantly impacts the performance of classification models, especially in the context of limited or complex datasets. This study aims to compare various augmentation techniques in classifying nitik batik motifs and provide new contributions to developing more effective classification methods for culture-based applications.

3. METHOD

The research uses a comparative approach. Several augmentation techniques are applied to the nitik batik motif dataset, and the results are compared based on classification performance metrics. Figure 1 shows the stages of the research methodology that were conducted in the research.

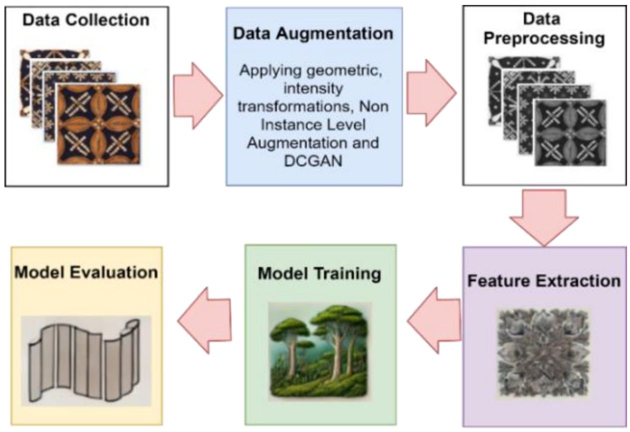


Figure 1. The conducted research methodology

3.1. Data collection

The public dataset contains images of nitik batik motifs [33]. This dataset consists of 240 images, consisting of 60 nitik batik motifs (equal number of images per category). Each image is 512×512 pixels in size. Figure 2 shows a sample of nitik batik motifs in the dataset.

Figure 2. A sample of nitik batik

3.2. Augmentation technique

In general, classification model including the classification of nitik batik, applying various data augmentation techniques is essential for enhancing model's performance. These methods introduce variability to the dataset, improving the model's generalization capabilities. This approach evaluates which augmentation technique provides the most effective results for accurately classifying nitik batik motifs while preserving the essential patterns.

The augmentation techniques were selected based on their ability to address the challenges of nitik batik classification, including high intraclass variability, complex dot-based patterns, and limited dataset size (240 images). Geometric transformations (flipping, rotation, scaling, shearing, and translation) were chosen to enhance model robustness to positional and orientational variations, as these are common in real-world batik images [14], [18], [19]. Intensity transformations (cut-out, grid mask, hide and seek, and random erasing) were selected to simulate occlusions and lighting variations, improving generalization to imperfect images [15], [25], [26]. Cut-out and random erasing, in particular, introduce local disruptions, mimicking real-world imperfections while preserving overall motif structure. Pairing samples was included to explore non-instance-level augmentation, combining images to create diverse hybrid patterns, as demonstrated in [34]. DCGAN was chosen to generate synthetic nitik batik images, addressing dataset limitations by producing high-quality samples that capture intricate textures [16], [17]. These methods were selected to balance diversity, robustness, and preservation of nitik batik's cultural and visual characteristics. Table 1 summarizes the augmentation techniques and their parameters, with a detailed description of the DCGAN architecture provided. A detailed description of the DCGAN architecture is provided in Figure 3, illustrating the layer configurations, dimensions, and data flow for both the generator and discriminator.

Table 1. Augmentation techniques and parameters

Augmentation	Technique	Description and parameters
Geometric transformations [35]	Rotation	Random rotation within [-30°, 30°], step size of 10°.
	Flipping	Horizontal and vertical flipping with a probability of 0.5.
	Scaling	Random scaling between 0.8× and 1.2× of original image size.
	Shearing	Shear angle range of [-15°, 15°].
	Translation	Random shifts along X and Y axes within [-10%, 10%] of image dimensions.
Intensity transformations [36]	Cut-out	Random removal of 1–3 square patches (50×50 pixels) per image.
	Grid mask	Grid of 10×10 pixel blocks, 50% probability of masking each block.
	Hide and seek	Randomly hide 16×16 pixel patches, covering 20% of the image.
	Random erasing data	Erase rectangular regions (10%–20% of image area) filled with random noise.
Non instance level augmentation [34]	Pairing samples	Combine two images with a blending ratio of 0.5 (equal contribution).
Unconditional image generation [37]	DCGAN	The generator takes a 100-dimensional noise vector and upsamples it through a series of transposed convolutional layers to produce 512×512 grayscale images, matching the preprocessed nitik batik dataset. The discriminator evaluates whether input images are real (from the 240-image dataset) or synthetic. The model was trained for 200 epochs with a batch size of 32, using the Adam optimizer (learning rate 0.0002, β1 =0.5, β2 =0.999) and binary cross-entropy loss.

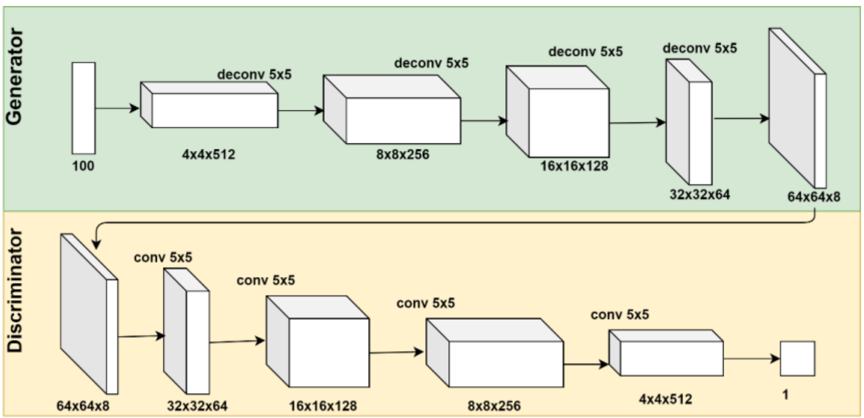


Figure 3. DCGAN architecture for generating synthetic nitik batik images

The DCGAN was implemented using PyTorch, following framework proposed by Radford *et al.* [38]. Training was conducted on the original 240 nitik batik images, generating 720 synthetic images. These images were visually inspected to ensure they preserved the characteristic dot-based geometric patterns of nitik batik motifs. The architecture was tuned to capture the intricate textures of nitik batik, with filter sizes and layer depths adjusted to handle the 512×512 resolution. All augmentation techniques were implemented using standard libraries. Parameters were selected based on established practices and tuned to balance diversity and preservation of nitik batik patterns, ensuring the augmented dataset enhances model generalization while maintaining cultural and visual integrity.

3.3. Pre-processing

The pre-processing used in our research converts the image into grayscale. This is due to the relatively limited color variations in nitik batik fabrics. By converting the image to grayscale, the focus shifts from color information, which is not critical in this case, to the intricate motifs and textures that define nitik batik. This simplification reduces data complexity and enhances the model's ability to extract meaningful features related to the patterns and structures in the batik [39], leading to more efficient and accurate classification results.

3.4. Feature extraction

We use the binarized statistical image features (BSIF) for feature extraction because it is a practical feature extraction method to capture batik's complex and unique textures and motifs. BSIF uses filters learned from natural images to extract local statistical features. This makes it very useful for analyzing complex patterns such as those in nitik batik.

Images were converted to grayscale to focus on texture patterns, as nitik batik motifs are defined by dot-based structures rather than color. Grayscale images were normalized to [0, 1] for consistent feature extraction. BSIF filters, inspired by independent component analysis (ICA) [40], were applied to extract texture features. These pre-trained filters, derived from natural images, use 8-bit, 5×5 patches to capture local texture patterns, ideal for encoding the edge-like and geometric features of nitik batik. Visual inspection confirmed that BSIF processing preserved critical dot-based patterns, and the high classification accuracy validates the dataset's suitability. The dataset's balanced structure (16 images per category) and high resolution ensure sufficient diversity and detail, making it appropriate for texture-based feature extraction without losing essential motif characteristics.

The formula for BSIF can be described as follows:

i) A local neighborhood patch represents each pixel in a grayscale image.
ii) A set of learned filters $W_1, W_2, \cdots, W_n$ is applied to these patches.
iii) The output of each filter is binarized using (1):

$$b_i = \begin{cases} 1 & \text{if } W_i^T \, patch \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

iv) The binary responses $b_1, b_2, \cdots, b_n$ form a binary string for each pixel, which is then used as the feature descriptor.

These binary codes efficiently capture local textures, which is particularly useful in recognizing the complex motifs in nitik batik classification.

3.5. Classification model development

Random forest is an effective classification model for nitik batik because it handles complex, high-dimensional data such as texture patterns. The model works by building multiple decision trees during training, each analyzing different features of the batik motif. This model builds multiple decision trees and combines their predictions for the classification of nitik batik. The critical formula used in random forest is [41]:

i) Gini impurity or entropy for classification trees can be seen at (2) and (3):

$$Gini(D) = 1 - \sum_{i=1} p_i^2 \tag{2}$$

$$Entropy(D) = -\sum_{i=1}^{n} p_i log_2(p_i) \tag{3}$$

Where p_i is the proportion of class i in the dataset D.

ii) Prediction: the final prediction for classification is the majority vote from all trees.

Random forest is well-suited to handle noise and variance in the data. It is a great choice for nitik batik, where classification depends on detailed texture information rather than primary pixel-based data. To validate the nitik batik classification model, we employ k-fold cross-validation, emphasizing the importance of data diversity and reliability. This approach divides the dataset into $k = 4$ folds of equal size. The model is trained on $(k - 1)$ folds and tested on the remaining folds. This process repeats k, ensuring that every sample is used for training and validation. By averaging the results, this method reduces bias and variance, thus providing a robust evaluation of the model's performance on different subsets of data.

3.6. Evaluation

Once the classification model has been developed, it was evaluated for assessing its accuracy to determine how well the model can classify different batik motifs. An additional important metric is the standard deviation of the accuracy across multiple cross-validation folds, as it measures the stability of the model. A lower standard deviation indicates consistent performance across different data subsets, implying that the model generalizes well. This combination of metrics ensures that the classification model is accurate and reliable for real-world application.

4. RESULTS AND DISCUSSION

After applying data augmentation techniques on nitik batik images using geometric transformation, intensity transformation, non-instance level, and unconditional image generation. Figure 4 illustrates the preprocessing outcomes for representative nitik batik images. Figure 4(a) shows an original RGB image of the 'Sekar cengkeh' motif, Figures 4(b) to 4(f) displays its geometric transformation augmentation, Figures 4(g) to 4(j) present the intensity transformation, Figure 4(k) depict the non-instance level augmentation, and Figure 4(l) shows unconditional image generation augmentation.

The results of flipping in Figure 4(b) and rotating in Figure 4(c) augmentations resemble the original. Both techniques appear to retain the overall pattern of the motif, only changing the orientation and direction. These transformations maintain the integrity of the original nitik batik design. The motifs that change the most from the original are the cut-out in Figure 4(g) and random erasing data in Figure 4(j) augmentations. These augmentations drastically modify the pattern by removing parts of the motif, thus changing its appearance significantly. Pairing samples in Figure 4(k) also dramatically alters the image by combining different motifs, creating a hybrid pattern.

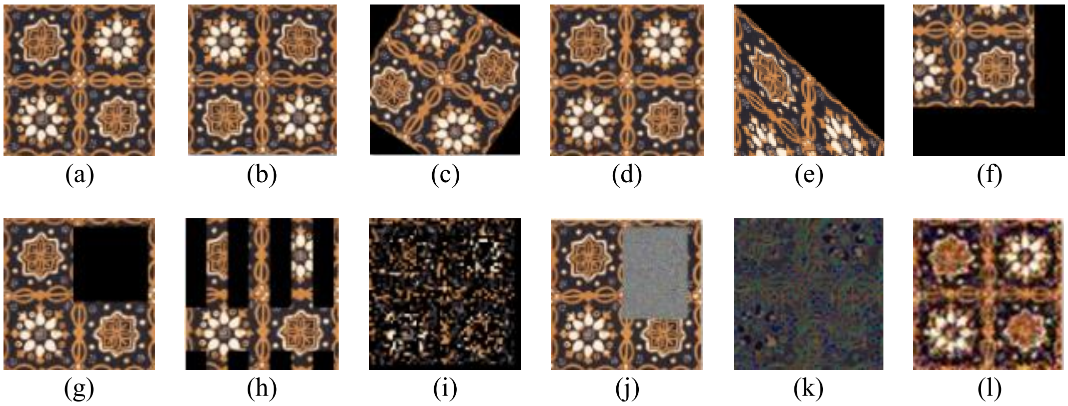


Figure 4. Sample result of data augmentation on the sekar cengkeh motif: (a) no augmentation, (b) flip, (c) rotate, (d) scaling, (e) shearing, (f) translation, (g) cut-out, (h) grid mask, (i) hide and seek, (j) random erasing data, (k) pairing samples, and (l) DCGAN

BSIF extracts texture features by applying learned filters to local image patches. We use the parameters filter size 15×15 and filter bit 10×10 to specify the texture extraction in detail. The extracted features are stored in a feature array and saved to a CSV file. Figure 5 shows the binary feature distribution to help assess the spread and intensity of the features in the form of a histogram. Figure 5(a) shows the binary feature distribution for original image, Figures 5(b) to 5(f) display shows the binary feature distribution for geometric transformation augmentation, Figures 5(g) to 5(j) depict the binary feature distribution for intensity transformation augmentation, Figure 5(k) present the binary feature distribution for non-instance level, and Figure 5(l) show the binary feature distribution for unconditional image generation.

Figure 5 presents the plot as a stacked histogram representing the distribution of pixel intensity values or feature values extracted from several images. Overlaid colors indicate different batches of processed images. The bell-shaped pattern indicates that most values cluster around the center, reflecting a balanced intensity distribution or concentration of features around a particular central value. The x-axis represents the range of intensity values that result from the convolution of the image with BSIF filters (from -4 to 4), indicating how well the texture in the image aligns with the learned filters. The values give insights into the strength of the texture patterns and their alignment with the filters. The y-axis on the histograms represents the frequency or count of pixel intensity values or feature responses extracted using the

BSIF method. The differences in y-axis values across the histograms occur because each augmentation technique alters the distribution and number of features or pixel values the BSIF filter captures.

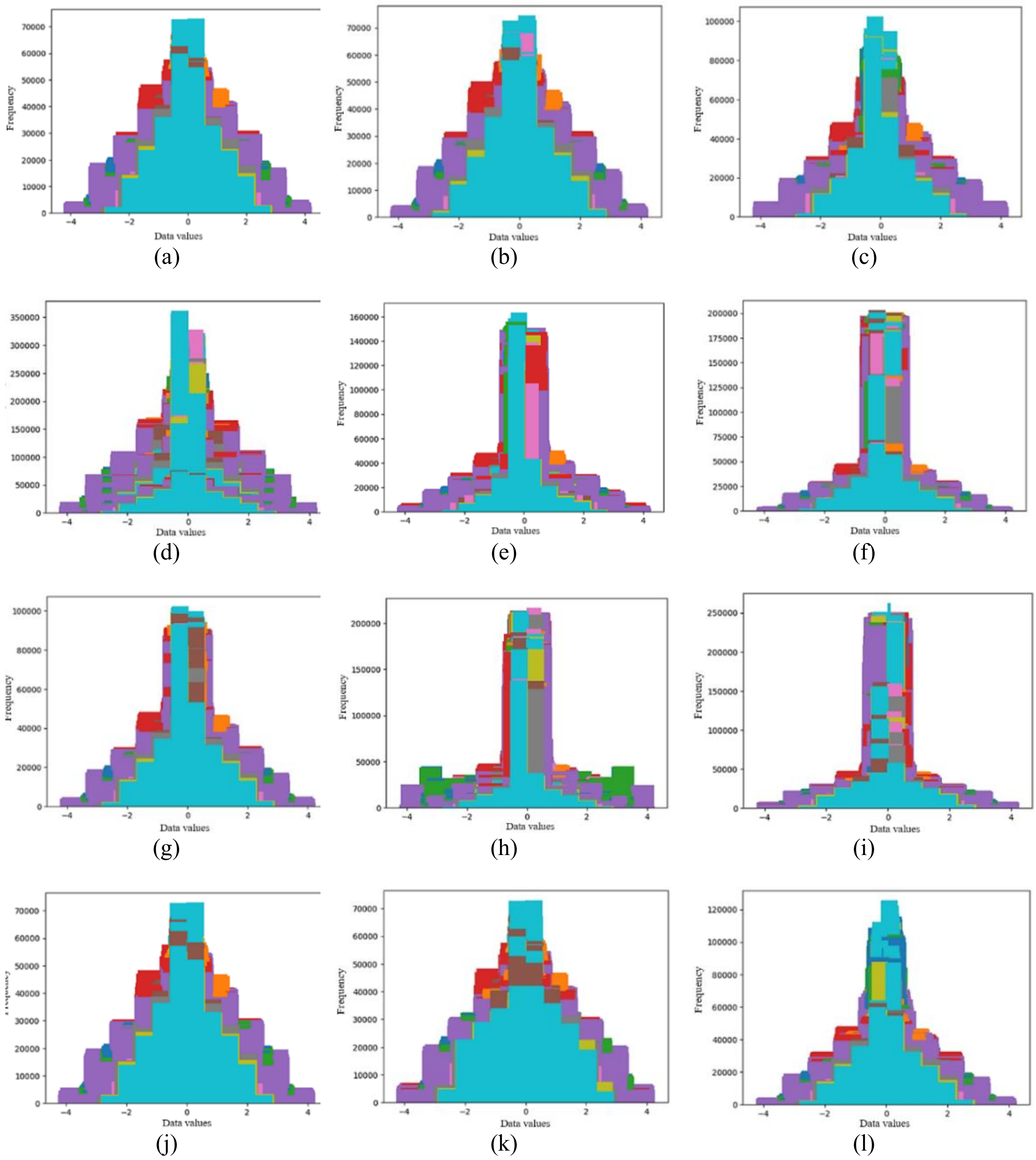


Figure 5. The binary feature distribution using BSIF: (a) no augmentation, (b) flip, (c) rotate, (d) scaling, (e) shearing, (f) translation, (g) cut-out, (h) grid mask, (i) hide and seek, (j) random erasing data, (k) pairing samples, and (l) DCGAN

Each augmentation method impacts the feature extraction process differently. The BSIF features extracted after flipping in Figure 5(b), the image may appear quite similar to the original (no augmentation), as flipping does not significantly alter the local texture but changes the orientation. Thus, the extracted texture features remain consistent. Rotation, scaling, and shearing in Figures 5(c) to 5(e) introduce noticeable changes to the features due to alterations in orientation and size. Cut-out, grid mask, and hide and seek in Figures 5(g) to 5(i) lead to sparse or segmented feature distributions due to occlusion. DCGAN in Figure 5(l) shows the impact of synthetic data generation on feature representation. Each augmentation method affects the stability and consistency of the extracted BSIF features, which directly influences the classification performance.

After extracting features from each augmented nitik batik image using BSIF, we evaluated the impact of various augmentation techniques on the accuracy and stability of a random forest-based classification model. Table 2 compares image augmentation techniques applied to the nitik batik classification model. It focuses on accuracy and stability (standard deviation) across four cross-validation folds.

Table 2. Comparison of image augmentation techniques applied to nitik batik classification

Augmentation		Cross validation (%)				Average	Standard
		1	2	3	4	accuracy (%)	deviation (%)
No augmentation		80.00	70.00	85.00	75.00	77.50	5.59
Geometric transformation	Flip	9.50	99.16	99.58	99.58	97.70	3.01
	Rotate	41.25	43.75	42.91	41.66	42.39	0.99
	Scaling	69.16	57.91	62.91	65.41	63.85	4.08
	Sharing	65.00	35.00	41.66	42.50	46.04	11.32
	Translation	74.58	61.66	70.41	71.66	69.58	4.81
Intensity transformation	Cut-out	99.58	95.83	95.41	96.25	96.77	1.65
	Grid mask	92.50	70.41	71.25	66.25	75.10	10.22
	Hide and seek	37.91	14.58	14.16	13.75	20.10	10.28
	Random erasing data	94.16	89.58	89.16	83.75	89.16	3.69
Non-instance level	Pairing samples	3.33	11.66	13.75	9.16	9.47	3.90
Unconditional image generation	DCGAN	93.75	92.50	94.58	94.16	93.75	0.78

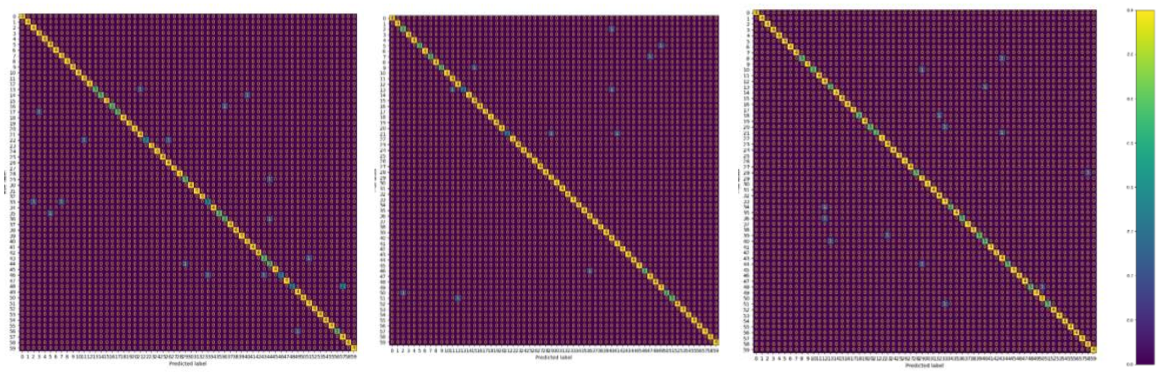
The evaluation of image augmentation techniques on the nitik batik 960 dataset revealed significant variations in classification performance using a random forest model, as shown in Table 2. Without augmentation, the model achieved an accuracy of 77.50% with a standard deviation of 5.59%, indicating reasonable performance but higher variance across folds. In contrast, augmentation methods like flipping (97.70% accuracy, 3.01% standard deviation), cut-out (96.77% accuracy, 1.65% standard deviation), and DCGAN (93.75% accuracy, 0.78% standard deviation) substantially improved both accuracy and stability. These results underscore the importance of augmentation in enhancing model performance on the limited and complex nitik batik dataset.

The superior performance of flipping, cut-out, and DCGAN can be attributed to their ability to balance dataset diversity with the preservation of nitik batik's intricate dot-based patterns. Flipping maintains geometric motif integrity by altering orientation without disrupting texture, enabling consistent feature extraction via BSIF. Cut-out introduces controlled occlusions, simulating real-world imperfections like fabric wear while retaining sufficient pattern information for robust classification. DCGAN, leveraging CNN architecture, generates high-quality synthetic images that closely mimic nitik motifs, as confirmed by visual inspection, providing diverse training samples with minimal variance (lowest standard deviation of 0.78%). Conversely, methods like pairing samples (9.47% accuracy), hide and seek (20.10% accuracy), and grid mask (75.10% accuracy) underperformed due to excessive disruption of texture coherence. Pairing samples creates hybrid patterns by combining unrelated motifs, confusing the classifier, while hide and seek and grid mask introduce occlusions that obscure critical dot patterns, highlighting the need for augmentation methods that preserve nitik batik's unique textures.

Random forest was selected as the classifier due to its alignment with the study's objective of evaluating augmentation effects on a small, high-dimensional dataset. Its robustness to BSIF-extracted texture features and effectiveness on the nitik batik 960 dataset (960 images across 60 motif categories) make it suitable for capturing intricate dot-based patterns, as evidenced by the high accuracies with flipping and cut-out. The integration of CNN architecture within DCGAN, which achieved strong accuracy (93.75%) and the highest stability, further supports random forest's representativeness. By leveraging CNN-generated synthetic images, random forest benefits from enhanced dataset diversity without requiring a CNN-based classifier, maintaining the study's focus on augmentation impacts. Comparing random forest to other classifiers like CNNs would necessitate extensive experiments and shift attention away from augmentation effects, which is beyond this study's scope.

Despite its strengths, random forest has limitations. Its reliance on handcrafted BSIF features may not capture multi-level hierarchical patterns as effectively as deep learning models like CNNs, which learn features directly from raw images. Additionally, its performance is sensitive to augmentation quality, as seen in the low accuracies of pairing samples and hide and seek, which disrupted motif coherence. Nevertheless, random forest's strong performance, interpretability, and alignment with the study’s goals make it an appropriate choice for assessing the impact of augmentation techniques on nitik batik classification.

We take the confusion matrix on three classification models that have high performance. The model was built with data using flipping, cut-out, and DCGAN augmentation. Classification results are presented in matrix format, where rows represent actual classes, and columns represent predicted classes. Diagonal elements indicate correct predictions, while off-diagonal elements indicate misclassification. Figure 6 visualizes the classification performance metrics. Figure 6(a) displays a confusion matrix for the flipping-augmented dataset. Figure 6(b) presents a confusion matrix for the cut out-augmented dataset. Figure 6(c) shows a confusion matrix for the DCGAN-augmented dataset. These results demonstrate that augmentation methods preserving texture coherence significantly enhance classification performance.



(a) (b) (c)

Figure 6. Confusion matrices of the classification model using different augmentation techniques: (a) flipping, (b) cut-out, and (c) DCGAN.

Figure 6 shows that each augmentation method has a slightly different effect on the model's performance, with flip and DCGAN showing the most consistent results. At the same time, cut-out may introduce more complexity due to image occlusion. The three confusion matrices show that five nitik batik motifs are incorrectly classified: sekar keben, sekar dangan, gedhangan, sekar pala, and sekar kenanga. Figure 7 shows the five nitik batik motifs that are incorrectly classified.

Sekar keben Sekar dangan Gedhangan Sekar pala Sekar kenanga

Figure 7. The five nitik batik motifs that were incorrectly classified

The batik motifs displayed (sekar keben, sekar dangan, gedhangan, sekar pala, and sekar kenanga) may be challenging to predict by the model due to several factors: i) the five nitik batik motifs have similar geometric shapes or repeating patterns (for example, sekar keben and sekar pala have circular and symmetrical elements), so they can confuse the model. If the features extracted by the model focus more on geometric structure than fine details, this can lead to misclassification; ii) subtle texture differences cause the texture differences to be subtle and not easily captured by the BSIF feature extraction method; and iii) all motifs have a similar color scheme (mainly dark backgrounds with lighter patterns). Because color information is less relevant or discarded entirely in grayscale feature extraction methods, the model may need help differentiating effectively based on texture alone.

5. CONCLUSION

This research has comprehensively evaluated various image data augmentation techniques in the context of nitik batik motif classification. By comparing geometric transformations, intensity transformations, non-instance level augmentation, and unconditional image generation, we identified their impact on the performance of random forest-based classification models. The results demonstrate that specific augmentation techniques, namely flipping, cut-out, and DCGAN, significantly enhance the accuracy of the nitik batik classification model. Flipping achieved the highest accuracy improvement of 20.20% compared to the baseline model without augmentation, followed by cut-out at 19.27% and DCGAN at 16.25%. Notably, the DCGAN augmentation technique exhibited the highest stability, with a standard deviation of 0.78%, indicating consistent performance across validation folds. The findings have important implications for both practical and theoretical domains. Practically, the improved classification accuracy can be used for the development of automated batik motif recognition systems, supporting cultural preservation efforts, museum digitization, and applications in the creative industry. Theoretically, this research demonstrates how different augmentation techniques can overcome the challenges associated with limited and complex datasets, providing insight into their effectiveness and stability in computer vision tasks. This research has some limitations. The dataset used is relatively small, consisting of only 240 images, which does not fully represent the diversity of nitik batik motifs in real-world scenarios. Moreover, the use of grayscale preprocessing may have overlooked potential color-based features that could improve the classification performance for specific motifs. Finally, although the random forest classifier proved to be effective, it may not fully exploit the capabilities of more advanced deep learning models, which could offer further improvements. For future research, we suggest exploring the incorporation of multiple augmentation techniques to enhance the model's robustness further. In addition, experimenting with larger and more diverse datasets, including images with varying illumination or occlusion, may enhance the model's generalizability. Lastly, evaluating the performance of deep learning architecture along with augmentation techniques may provide further insights to optimize the classification of nitik batik.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Suprapto	✓	✓	✓	✓	✓	✓		✓		✓		✓	✓	
Meilany Nonsi Tentua		✓				✓		✓	✓	✓	✓			
Ahmad Rizki Maulana	✓		✓	✓			✓			✓	✓			

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in Mendeley Data at http://doi.org/10.17632/sgh484jxzy.3.

REFERENCES

[1] A. D. Wibawa, E. A. Wicaksono, S. D. Suryani, and R. Rumadi, "Javanese batik image classification using self-organizing map," in *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, 2023, pp. 472–477, doi: 10.1109/ICCoSITE57641.2023.10127783.

[2] J. Kusanti, E. Noersasongko, M. A. Soeleman, F. Alzami, Purwanto, and Z. A. Hasibuan, "Pre-processing of resize and region of interest (ROI) to improve the accuracy of batik detection based on VGG-16," in *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2023, pp. 242–247, doi: 10.1109/EECSI59885.2023.10295666.

[3] A. R. Zuhro, I. K. Sunarya, and W. Nugraheni, "Batik nitik's existence in the postmodern era," in *Proceedings of the 3rd International Conference on Arts and Arts Education (ICAAE 2019)*, 2020, pp. 1–6, doi: 10.2991/assehr.k.200703.001.

[4] A. R. Maulana, Suprapto, and D. A. Tyas, "Classification of batik motifs using multi-texton co-occurrence descriptor and binarized statistical image features," *Journal of Applied Data Sciences*, vol. 6, no. 2, pp. 1367–1378, 2025, doi: 10.47738/jads.v6i2.622.

[5] I. Setyawan, I. K. Timotius, and M. Kalvin, "Automatic batik motifs classification using various combinations of SIFT features moments and k-nearest neighbor," in *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2015, pp. 269–274, doi: 10.1109/ICITEED.2015.7408954.

[6] A. E. Minarno, Y. Azhar, F. D. S. Sumadi, and Y. Munarko, "A robust batik image classification using multi texton co-occurrence descriptor and support vector machine," in *2020 3rd International Conference on Intelligent Autonomous Systems (ICoIAS)*, 2020, pp. 51–55, doi: 10.1109/ICoIAS49312.2020.9081833.

[7] C. S. K. Aditya, M. Hani'ah, R. R. Bintana, and N. Suciati, "Batik classification using neural network with gray level co-occurence matrix and statistical color feature extraction," in *2015 International Conference on Information & Communication Technology and Systems (ICTS)*, 2015, pp. 163–168, doi: 10.1109/ICTS.2015.7379892.

[8] A. Tejawati *et al.*, "Batik pattern classification using decision tree based on color-texture feature," in *2023 Sixth International Conference on Vocational Education and Electrical Engineering (ICVEE)*, 2023, pp. 347–352, doi: 10.1109/ICVEE59738.2023.10348243.

[9] A. E. Minarno, I. Soesanti, and H. A. Nugroho, "Batik image retrieval using convolutional autoencoder," in *2024 IEEE 14th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2024, pp. 15–20, doi: 10.1109/ISCAIE61308.2024.10576476.

[10] S. Mohammed *et al.*, "The effects of data quality on machine learning performance on tabular data," *Information Systems*, vol. 132, 2025, doi: 10.1016/j.is.2025.102549.

[11] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," *Pattern Recognition*, vol. 137, 2023, doi: 10.1016/j.patcog.2023.109347.

[12] A. Mumuni and F. Mumuni, "Data augmentation: a comprehensive survey of modern approaches," *Array*, vol. 16, 2022, doi: 10.1016/j.array.2022.100258.

[13] Y. Zhou, C. Guo, X. Wang, Y. Chang, and Y. Wu, "A survey on data augmentation in large model era," *arXiv-Computer Science*, pp. 1–33, 2024.

[14] I. B. K. Manuaba and V. J. Basiroen, "Comparison study of image augmentation on modified CNN architecture for Indonesian Lasem-batik's motifs," *ICIC Express Letters*, vol. 17, no. 4, pp. 409–417, 2023, doi: 10.24507/icicel.17.04.409.

[15] M. A. Rasyidi, R. Handayani, and F. Aziz, "Identification of batik making method from images using convolutional neural network with limited amount of data," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 3, pp. 1300–1307, 2021, doi: 10.11591/eei.v10i3.3035.

[16] M. Abdurrahman, N. H. Shabrina, and D. K. Halim, "Generative adversarial network implementation for batik motif synthesis," in *2019 5th International Conference on New Media Studies (CONMEDIA)*, 2019, pp. 63–67, doi: 10.1109/CONMEDIA46929.2019.8981834.

[17] A. E. Minarno, M. C. Mustaqim, Y. Azhar, W. A. Kusuma, and Y. Munarko, "Deep convolutional generative adversarial network application in batik pattern generator," in *2021 9th International Conference on Information and Communication Technology (ICoICT)*, 2021, pp. 54–59, doi: 10.1109/ICoICT52021.2021.9527514.

[18] S. Aras, A. Setyanto, and Rismayani, "Classification of Papuan batik motifs using deep learning and data augmentation," in *2022 4th International Conference on Cybernetics and Intelligent System (ICORIS)*, 2022, pp. 1–5, doi: 10.1109/ICORIS56080.2022.10031320.

[19] D. Trimakno and Kusrini, "Impact of augmentation on batik classification using convolution neural network and k-neareast neighbor," in *2021 4th International Conference on Information and Communications Technology (ICOIACT)*, 2021, pp. 285–289, doi: 10.1109/ICOIACT53268.2021.9564000.

[20] B. S. Negara, E. Satria, S. Sanjaya, and D. R. D. Santoso, "ResNet-50 for classifying Indonesian batik with data augmentation," in *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, 2021, pp. 1–4, doi: 10.1109/ICOTEN52080.2021.9493488.

[21] I. M. A. Agastya and A. Setyanto, "Classification of Indonesian batik using deep learning techniques and data augmentation," in *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*, 2018, pp. 27–31, doi: 10.1109/ICITISEE.2018.8720990.

[22] Farida, R. E. Caraka, T. W. Cenggoro, and B. Pardamean, "Batik parang rusak detection using geometric invariant moment," in *2018 Indonesian Association for Pattern Recognition International Conference (INAPR)*, 2018, pp. 71–74, doi: 10.1109/INAPR.2018.8627000.

[23] F. A. Putra *et al.*, "Classification of batik authenticity using convolutional neural network algorithm with transfer learning method," in *2021 Sixth International Conference on Informatics and Computing (ICIC)*, 2021, pp. 1–6, doi: 10.1109/ICIC54025.2021.9632937.

[24] R. G. Tiwari, A. K. Agarwal, V. Jain, and A. Kumar, "Batik classification in Indonesia: exploring its significance on tourism and economy," in *2023 International Conference on Sustaining Heritage: Innovative and Digital Approaches (ICSH)*, 2023, pp. 119–124, doi: 10.1109/ICSH57060.2023.10482828.

[25] C. U. Khasanah, E. Utami, and S. Raharjo, "Implementation of data augmentation using convolutional neural network for batik classification," in *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, 2020, pp. 1–5, doi: 10.1109/CITSM50537.2020.9268890.

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

[27] A. Kitipong, W. Rueangsirasak, and R. Chaisricharoen, "Classification system for traditional textile: case study of the batik," in *2013 13th International Symposium on Communications and Information Technologies (ISCIT)*, 2013, pp. 767–771, doi: 10.1109/ISCIT.2013.6645958.

[28] N. Suciati, A. Kridanto, M. F. Naufal, M. Machmud, and A. Y. Wicaksono, "Fast discrete curvelet transform and HSV color features for batik image clansificotlon," in *2015 International Conference on Information & Communication Technology and Systems (ICTS)*, 2015, pp. 99–104, doi: 10.1109/ICTS.2015.7379879.

[29] N. Suciati, W. A. Pratomo, and D. Purwitasari, "Batik motif classification using color-texture-based feature extraction and backpropagation neural network," in *2014 IIAI 3rd International Conference on Advanced Applied Informatics*, 2014, pp. 517–521, doi: 10.1109/IIAI-AAI.2014.108.

[30] A. E. Minarno, I. Soesanti, and H. A. Nugroho, “A convolutional neural network model for batik image retrieval,” in *2024 IEEE 14th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2024, pp. 31–36, doi: 10.1109/ISCAIE61308.2024.10576422.

[31] B. H. Lifindra, D. Herumurti, and A. Yuniarti, “A comparison of VGG architecture convolutional layers in migrating batik style into fractal shape,” in *2024 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, 2024, pp. 268–273, doi: 10.1109/SIML61815.2024.10578183.

[32] A. Fadlil, I. Riadi, and I. J. D. E. P. Putra, “Comparison of machine learning performance using naive Bayes and random forest methods to classify batik fabric patterns,” *Revue d'Intelligence Artificielle*, vol. 37, no. 2, pp. 379–385, 2023, doi: 10.18280/ria.370214.

[33] A. E. Minarno, I. Soesanti, and H. A. Nugroho, “Batik nitik 960 dataset for classification, retrieval, and generator,” *Data*, vol. 8, no. 4, 2023, doi: 10.3390/data8040063.

[34] H. Inoue, “Data augmentation by pairing samples for images classification,” *arXiv-Computer Science*, pp. 1–8, 2018.

[35] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[36] H. Naveed, S. Anwar, M. Hayat, K. Javed, and A. Mian, “Survey: image mixing and deleting for data augmentation,” *Engineering Applications of Artificial Intelligence*, vol. 131, 2024, doi: 10.1016/j.engappai.2023.107791.

[37] Y. Lu, D. Chen, E. Olaniyi, and Y. Huang, “Generative adversarial networks (GANs) for image augmentation in agriculture: a systematic review,” *Computers and Electronics in Agriculture*, vol. 200, 2022, doi: 10.1016/j.compag.2022.107208.

[38] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016, pp. 1–16.

[39] W. Liu, N. Lv, J. Wan, L. Wang, and X. Zhou, “Pixel embedding for grayscale medical image classification,” *Heliyon*, vol. 10, no. 16, 2024, doi: 10.1016/j.heliyon.2024.e36191.

[40] J. Kannala and E. Rahtu, “BSIF: Binarized statistical image features,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012, pp. 1363–1366.

[41] E. Xi, “Image classification and recognition based on deep learning and random forest algorithm,” *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, 2022, doi: 10.1155/2022/2013181.

BIOGRAPHIES OF AUTHORS

Suprapto received a B.S. degree in Mathematics from Universitas Gadjah Mada, Indonesia, in 1987. He received his M.S. degree in computer science from Universitas Indonesia, Indonesia, and a Doctorate in Computer Science from Universitas Gadjah Mada, Indonesia, in 2015. From 2001, he was a lecturer at the Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia. Since 2024, he has been a Professor at the Department of Computer Science and Electronics, Gadjah Mada University, Indonesia. His research interests include simulation modeling and computational modeling. He is a recipient of a sandwich program at UI-UMUC, Maryland, USA, in 1990. Since 2021, he has served as chairman of the Certification Department in the IndoCEISS. Since 2015, he has been registered as a member of Aptikom and IAENG. He can be contacted at email: sprapto@ugm.ac.id.

Meilany Nonsi Tentua received a B.S. degree in Mathematics from Universitas Gadjah Mada, Indonesia, in 1996. She received a Master's Degree in Electrical Engineering from Universitas Gadjah Mada, Indonesia, in 2004, and a Doctorate in Computer Science from Universitas Gadjah Mada, Indonesia, in 2023. She has been a lecturer at Informatics, Universitas PGRI Yogyakarta, Indonesia, since 2006. Her research interests include machine learning, data science, data mining, natural language processing, and artificial intelligence. She can be contacted at email: meilany@upy.ac.id.

Ahmad Rizki Maulana received a B.S. degree in Informatics from Universitas Negeri Surabaya, Indonesia, in 2021. He is currently pursuing an M.S. degree in Computer Science at Universitas Gadjah Mada, Indonesia. His research interests include machine learning, data science, computer vision, and natural language. He can be contacted at email: ahmadrizkimaulana@mail.ugm.ac.id.