J. Ris. & Ap. Mat. Vol. 10, No. 1 (2026), pp. 48–58.
Jurnal Riset dan Aplikasi Matematika (JRAM)
e-ISSN: 2581-0154
DOI: doi.org/10.26740/jram.v10n1.p48-58

GENERATIVE CONVOLUTIONAL NEURAL NETWORK LEARNING WITH SIMPLE DENOISING ON GENERATING BATIK IMAGES

Maulida Yanti* and Putri K Nasution

Department of Mathematics, Universitas Sumatera Utara, Medan, Indonesia

*Corresponding Author: maulidayanti@usu.ac.id

ABSTRACT

Batik is a traditional textile from Indonesia, and automatic batik pattern generation can help enrich its diverse motifs. This research explores the use of a generative convolutional neural network, a generative random field model that learns its own filters directly from training data without relying on pre-trained features. The model parameters are estimated by minimizing the difference between the synthesized images and the training images. To improve image quality, k-means clustering and block-matching and three-dimensional (BM3D) filtering denoising are incorporated to reduce noise and enhance the Fréchet Inception Distance score. Two experiments are conducted. The first evaluates the model's ability to generate batik patterns from single input images using both high-quality (ITB-mBatik) and lower-quality internet-sourced datasets. The second examines the generation of blended batik patterns from pairs of internet-sourced batik images with complementary color and texture features. The results show that applying a simple denoising step improves the Fréchet Inception Distance score for the high-quality ITB-mBatik images, while showing little to no benefit for the lower-quality internet-sourced batik images. In addition, the model is able to generate visually appealing blended patterns, particularly when the input images share complementary color and texture features. These findings highlight the role of data quality and post-processing in generative batik pattern synthesis.

Keywords: *Generative Convolutional Neural Network, Random Field, Blended Batik Pattern, K-means Denoising, and BM3D Denoising*

1. Introduction

Batik is one of the most iconic traditional textiles of Indonesia, characterized by rich and diverse motifs that vary across regions. Efforts to preserve these cultural patterns while enabling the creation of new designs have motivated research in automatic batik pattern generation. Prior studies on batik image analysis and synthesis have evolved from classical pattern recognition to modern deep learning approaches. Early work, for example, relied on handcrafted features; Azhar *et al*. [1] applied support vector machines with SIFT descriptors for batik classification, while Situngkir [2] explored generative batik design using an L-system and the Thue–Morse algorithm.

More recently, deep generative models have significantly improved the ability to synthesize realistic batik motifs. Generative adversarial networks (GANs) [3, 4, 5, 6, 7] and denoising diffusion probabilistic models [8] have shown strong performance. However, these approaches are primarily based on adversarial or diffusion frameworks, while energy-based generative

2020 Mathematics Subject Classification: 68T07.
Received: 06-01-2026, revised: 17-04-2026, accepted: 18-04-2026.

models, such as generative convolutional neural network (ConvNet) [9], remain relatively less explored, particularly for structured texture synthesis like batik patterns.

Generative ConvNet originate from energy-based models and are closely related to the FRAME (Filters, Random field, and Maximum Entropy) model introduced by Zhu *et al*. [10]. This framework was later extended by Lu *et al*. [11] using pre-learned filters, and further developed by Xie *et al*. [9] into a fully trainable generative ConvNet that learns both filters and model parameters without relying on pre-trained weights. Under a Gaussian white noise reference, the model represents images as structured, piecewise Gaussian patterns and can also be interpreted in an autoencoder-like manner for image synthesis [12].

An important aspect of generative ConvNet is the presence of noise introduced during the learning and sampling process. In particular, Langevin dynamics injects Gaussian noise at each iteration to facilitate exploration of the image distribution. While necessary, this process can introduce high-frequency artifacts and minor structural inconsistencies in the generated images. In batik pattern synthesis, such noise may reduce the clarity of fine texture details and affect quantitative measures such as the Fréchet Inception Distance (FID). To reduce these effects, post-processing through denoising can be applied to improve the FID score of the synthesized batik patterns after denoising.

Motivated by these considerations, this study not only investigates the capability of generative ConvNet to synthesize batik patterns from single input images and to generate blended patterns from pairs of images, but also examines how noise and post-processing affect visual quality and FID scores. The model is applied to both high-quality and low-quality batik datasets, followed by post-processing using k-means and BM3D denoising methods. The study further analyzes whether denoising improves image quality or alters underlying texture characteristics, and how noise introduced during learning influences the resulting patterns and their FID scores.

2. Methodology

This section presents the theoretical background of the methodology used in this study. It includes the mathematical formulation of the generative ConvNet model, learning procedure for parameter estimation, and denoising methods applied to enhance the quality of the synthesized batik images.

2.1. Generative ConvNet Model

Following notation in [9], let an image defined on a rectangular (or square) domain $\mathcal{D}$ be denoted by $\mathbf{I}(x)$, where $x = (x_1, x_2)$ represents the pixel coordinates. The image $\mathbf{I}(x)$ can be considered either as a two-dimensional function or as a vector with a fixed pixel ordering [9]. Let the filtered image be denoted by $F * \mathbf{I}$, and the filter response at position x be written as $F * \mathbf{I}$, for a given filter F. In [9], the generative ConvNet is defined recursively, consisting of multiple layers of linear filtering followed by a Rectified Linear Unit (ReLU) as a nonlinear transformation.

Following [9], the generative ConvNet applies filtering and non-linear activation recursively across layers. The response at position x from filter $F_k^{(l)}$ at layer l is:

$$\left[F_k^{(l)} * \mathbf{I}\right](x) = h\left(\sum_{i=1}^{N_{l-1}} \sum_{y \in S_l} w_{i,y}^{(l,k)} \left[F_i^{(l-1)} * \mathbf{I}\right](x+y) + b_{l,k}\right), \tag{1}$$

or, in vector form:

$$\left[F_k^{(l)} * \mathbf{I}\right](x) = h\left(\left\langle \mathbf{w}_{k,x}^{(l)}, \mathbf{F}^{(l-1)} * \mathbf{I} \right\rangle + b_{l,k}\right), \tag{2}$$

where:

$F_k^{(l)} * \mathbf{I}$	:	the filtered image using filter k at layer l
$F_k^{(l)}$	:	the k^{th} filter at layer l, where $k \in \{1, 2, \ldots, N_l\}$
$h(\cdot)$	:	the Rectified Linear Unit (ReLU), defined as $h(r) = \max(r, 0)$
N_l	:	the number of filters at layer l
l	:	the index of the layer, where $l \in \{1, 2, \ldots, \mathcal{L}\}$
S_l	:	the spatial support of filters at layer l
$w_{i,y}^{(l,k)}$	:	the weight of filter k at layer l, applied to input channel i at position y
$b_{l,k}$	:	the bias term of filter k at layer l
$\mathbf{F}^{(l)} * \mathbf{I}$	:	a 3D feature map consisting of all N_l filtered outputs at layer l
$\mathbf{w}_{k,x}^{(l)}$	:	a locally supported 3D basis function aligned with $\mathbf{F}^{(l-1)} * \mathbf{I}$

Let $f(\mathbf{I}; w)$ and $q(\mathbf{I})$ be the scoring function and the Gaussian white noise distribution, respectively:

$$f(\mathbf{I}; w) = \sum_{k=1}^{K} \sum_{x \in \mathcal{D}_l} \left[F_k^{(l)} * \mathbf{I} \right](x), \tag{3}$$

$$q(\mathbf{I}) = \frac{1}{(2\pi\sigma^2)^{|\mathcal{D}|/2}} \exp\left[-\frac{1}{2\sigma^2} \|\mathbf{I}\|^2 \right], \tag{4}$$

where $|\mathcal{D}|$ is the number of pixels in the domain $\mathcal{D}$. Then the convolutional version of the generative ConvNet, defined by [9] as a Markov random field model, can be written as:

$$p(\mathbf{I}; w) = \frac{1}{Z(w)} \exp\left[f(\mathbf{I}; w) \right] q(\mathbf{I}), \tag{5}$$

where w includes the weights and bias terms that determine the filters ($F_k^{(l)},\ k = 1, \ldots, K = N_l$), $Z(w) = E\{\exp[f(\mathbf{I}; w)]\}$, $f(\mathbf{I}; w)$ is the scoring function, and $q(\mathbf{I})$ is the Gaussian white noise distribution.

Sampling from this model is carried out using Langevin dynamics:

$$\mathbf{I}_{t+1} = \mathbf{I}_t - \frac{\epsilon^2}{2} \left[\mathbf{I}_t - \mathbf{B}_{w,\delta(\mathbf{I}_t; w)} \right] + \epsilon Z_t, \tag{6}$$

where t and ϵ represent the time step and step size, respectively, and $Z_t \sim \mathcal{N}(0, 1)$. The term $\mathbf{B}_{w,\delta}$ is computed via the top-down deconvolution process:

$$\mathbf{B}^{(l-1)} = \sum_{k=1}^{N_l} \sum_{x \in \mathcal{D}_l} \mathbf{B}_k^{(l)}(x) \delta_{k,x}^{(l)}(\mathbf{I}; w) \mathbf{w}_{k,x}^{(l)}, \tag{7}$$

where $\mathbf{B}^{(l)} = (B_k^{(l)}(x),\ k = 1, \ldots, N_l,\ x \in \mathcal{D}_l)$, and $\mathbf{B}^{(\mathcal{L})}(x) = 1$ for all $k = 1, \ldots, N_{\mathcal{L}}$ and $x \in \mathcal{D}_{\mathcal{L}}$.

The activation pattern at each layer, denoted $\delta(\mathbf{I}; w) = (\delta_{k,x}^{(l)}(\mathbf{I}; w),\ \forall k, x, l)$, is computed in the bottom-up process using:

$$\delta_{k,x}^{(l)}(\mathbf{I}; w) = 1\left(\left\langle \mathbf{w}_{k,x}^{(l)}, \mathbf{F}^{(l-1)} * \mathbf{I} \right\rangle + b_{l,k} > 0 \right), \tag{8}$$

$$\left[F_k^{(l)} * \mathbf{I} \right](x) = \delta_{k,x}^{(l)}(\mathbf{I}; w) \left(\left\langle \mathbf{w}_{k,x}^{(l)}, \mathbf{F}^{(l-1)} * \mathbf{I} \right\rangle + b_{l,k} \right), \tag{9}$$

where $\mathbf{w}_{k,x}^{(1)}$ is a Gabor wavelet centered at position x, and $1(\cdot)$ is the indicator function. For the input layer, $\mathbf{F}^{(0)} * \mathbf{I}$ is defined as $\mathbf{I}$.

This formulation enables the generative ConvNet to define an energy-based probability model over natural images, making it particularly suitable for synthesizing structured textures such as batik patterns.

2.2. Learning Procedure for Parameter Estimation

The generative ConvNet is trained following the algorithm proposed in [9], which is used to estimate the parameters w and to generate synthesized images. Given a set of training images $\mathbf{I}_m$, where $m = 1, \ldots, M$ and M denotes the number of training samples, the weights w are estimated by maximizing the following average log-likelihood function:

$$L(w) = \frac{1}{M} \sum_{m=1}^{M} \log p(\mathbf{I}_m; w), \tag{10}$$

where $p(\mathbf{I}; w)$ is defined in Equation (5)

Following [9], the gradient of the log-likelihood with respect to the parameters w is given by:

$$\frac{\partial L(w)}{\partial w} = \frac{1}{M} \sum_{m=1}^{M} \frac{\partial}{\partial w} f(\mathbf{I}_m; w) - \mathbb{E}_w \left[\frac{\partial}{\partial w} f(\mathbf{I}; w) \right]. \tag{11}$$

To approximate the expectation term, the synthesized images $\tilde{\mathbf{I}}_m$ are updated by running Langevin dynamics (Equation (6)) for L steps. These synthesized samples serve as Monte Carlo approximations. The parameters w are then refined by computing the difference between the gradients obtained from the training data and those from the synthesized images. The overall learning procedure follows the iterative framework proposed by Xie *et al*. [9].

2.3. Denoising Methods

The synthesized batik images produced by generative ConvNet model often contain noise that degrades visual quality and affects quantitative evaluation. This noise is closely related to the learning and sampling process of the generative ConvNet. During training, the model learns the image distribution using Langevin dynamics, where Gaussian noise is injected at each iteration to enable exploration of the image space [9]. As a consequence, the generated images contain stochastic noise in the form of high-frequency pixel-level variations and small structural inconsistencies, particularly in homogeneous regions and smooth color areas.

This noise may affect the research results in two ways. First, it may reduce visual quality by introducing random pixel noise and irregular textures that are not present in real batik patterns. Second, it may affect the Fréchet Inception Distance (FID) by altering the feature statistics of the synthesized images, leading to greater differences between generated and real images. Consequently, this may result in higher FID scores.

To reduce this effect, post-processing via denoising is applied to the synthesized images. In this study, two denoising methods are considered: k-means clustering and Block-Matching and 3D Filtering (BM3D), applied individually and in combination. K-means denoising simplifies the image by clustering pixel colors into a fixed number of groups [13]. This reduces color variations caused by noise and produces smoother regions. However, since it operates only in the color space, it is less effective at removing random pixel noise and preserving structural details. BM3D denoising helps address this limitation by operating in both spatial and transform domains. It identifies similar image patches, groups them into 3D stacks, and applies collaborative filtering. As a result, BM3D can reduce noise while better preserving edges and texture details [14].

To systematically evaluate the effect of noise and denoising, FID scores are computed for both the original synthesized images (before denoising) and the denoised images. Three denoising scenarios are considered: (1) k-means denoising alone, (2) BM3D denoising alone, and (3) a sequential combination of k-means followed by BM3D. By first reducing color variation with k-means and then removing fine spatial noise using BM3D, this two-stage approach is expected to produce smoother and more structurally consistent batik patterns. The comparison between FID values before and after denoising provides a clearer assessment of how noise influences the evaluation and how effectively each method improves both visual quality and statistical similarity to real batik images.

3. Experimental Setup

3.1. Dataset

This research utilizes two data sources: (1) thirty clean, high-quality images randomly selected from the ITB-mBatik dataset (Figure 1), obtained from publicly available sources [15], and (2) modern batik images collected from the internet using search queries such as ”modern batik pattern”, which generally have lower pixel clarity compared to the ITB-mBatik dataset. Although approximately thirty modern batik images were initially gathered from online sources, only three images (Figure 2), are selected for further use due to their complementary color and texture features. This selection ensures that the patterns are compatible for blended pattern synthesis.

Both datasets are used to evaluate the model’s performance in generating batik patterns from single input images, by comparing high-quality inputs from the ITB-mBatik dataset with lower-quality internet-sourced images. This allows us to examine whether lower-quality inputs affect the model’s ability to generate visually coherent and realistic batik patterns. In addition, the three selected internet-sourced images are specifically used for the blended batik generation task to produce new motifs.

Although the number of training images is limited, the dataset consists of patterns with rich structural details. This study adopts a controlled experimental setting to examine the behavior of the generative model. This approach is consistent with common practices in texture synthesis, where representative patterns can be effectively learned from a relatively small number of samples.

Figure 1. Thirty batik images from the ITB-mBatik dataset.

(a) (b) (c)

Figure 2. Internet-sourced batik images

3.2. Experimental Framework

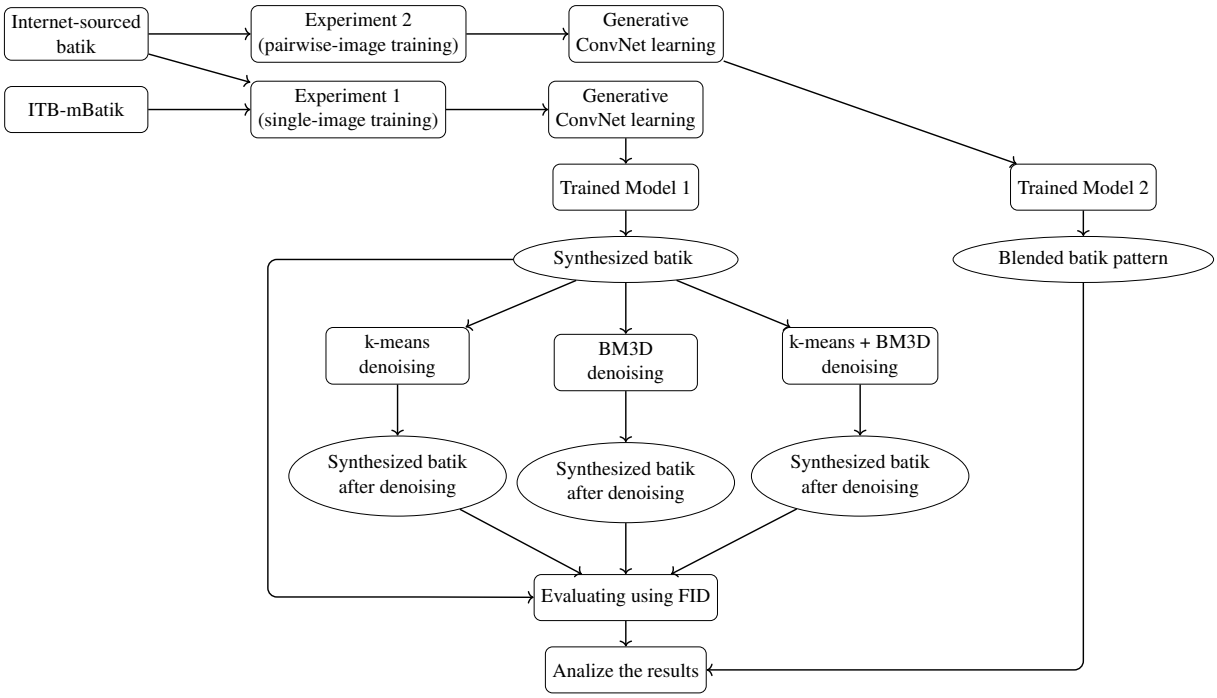


Figure 3. Flowchart of the experimental framework showing the workflow from input data and experimental setup to generative ConvNet training, image generation, denoising with three strategies, and evaluation using FID.

Figure 3 presents the overall workflow of the experimental framework. The process begins with two input sources, namely the ITB-mBatik dataset and internet-sourced batik dataset. These data are used in two experimental settings of the generative ConvNet model: Experiment 1 (single-image training) using both datasets, and Experiment 2 (pairwise-image training) using only the internet-sourced batik images.

In Experiment 1, the model is trained on a single image to evaluate its ability to generate batik patterns. The generative ConvNet is trained separately on the ITB-mBatik dataset (Figure 1) and the internet-sourced batik images (Figure 2). In Experiment 2, the model is trained on pairs of images to generate blended batik patterns, where the selected pairs share similar color and texture characteristics. Specifically, Experiment 2 uses two image pairs: (a) and (c), and (b) and (c) in Figure 2.

During training (generative ConvNet learning), the model is initialized and optimized through an iterative process that alternates between image synthesis and parameter updates. The model learns the image distribution defined by $p(I;w)$ in Eq. (5). At each iteration, synthesized images are generated using Langevin dynamics (Eq. (6)), and the model parameters are updated by matching statistics between real and synthesized images through the gradient of the log-likelihood (Eq. (11)), which depends on the score function $f(I;w)$ in Eq. (3). Gaussian noise

is injected at each step to facilitate exploration of the image distribution, although it may also introduce noise into the generated images.

Both experimental setting are implemented using the original code from [9] within the MatConvNet framework [16]. The generative ConvNet is trained in a layer-wise manner, where filters in the lower layers are refined via backpropagation as new layers were progressively added. Let $\tilde{M}$, L, and T denote the number of synthesized images used during Langevin sampling, the number of Langevin steps between consecutive parameter updates, and the number of learning iterations for each layer, respectively, following [9].

In Experiment 1, the model is trained with $\tilde{M} = 1$, $L = 3$, $T = 4000$, and a Gaussian reference distribution variance of $\sigma^2 = 1$. The architecture consisted of three convolutional layers: 150 filters of size 15×15 in the first layer, 64 filters of size 5×5 in the second layer, and 30 filters of size 3×3 in the third layer. Sub-sampling rates are set to 3 for the first layer and 1 for the remaining layers, and all layers are trained simultaneously using a one-step contrastive divergence method. In Experiment 2, the model is trained with $\tilde{M} = 2$, $L = 10$, and $T = 700$ for each layer, using the same filter configuration and sub-sampling rates. In this case, each layer was added sequentially, and no contrastive divergence step was applied during training.

The trained models are then used to generate new images. In Experiment 1, the model produces synthesized batik images, while in Experiment 2 it generates blended batik patterns. As described earlier, the generated images may contain noise due to the injection of Gaussian noise during the iterative sampling process. This noise can affect the synthesized batik patterns by degrading fine texture details, introducing random pixel noise, and increasing the difference between synthesized and real images.

To reduce these effects, three denoising methods are applied to the synthesized batik images: k-means clustering, BM3D filtering, and a combination of both. This post-processing step is introduced in this study as an additional enhancement and is not part of the original generative ConvNet implementation in [9]. The resulting images are then evaluated using the Fréchet Inception Distance (FID) to assess their similarity to real batik images, and the results are further analyzed to compare the performance across different denoising methods. This framework enables a systematic analysis of how denoising affects both visual quality and statistical similarity.

4. Result and Discussion

Figure 4 and Figure 5 present the synthesized batik images generated in Experiment 1 before applying any denoising method. The patterns derived from the ITB-mBatik dataset (Figure 4) closely resemble the training images, indicating that the model successfully captures the motifs and reproduces them with similar structure and color distribution. Similarly, the images generated from the internet-sourced batik dataset (Figure 5) also show visual consistency with their references. Although the ITB-mBatik dataset contains higher-quality images and the internet-sourced images have lower quality, the synthesized images from both datasets still contain slight noise, which appears as small dot-like patterns (random pixel noise) in the images.

To reduce this noise, three post-processing (denoising) strategies are applied: (1) k-means denoising, (2) BM3D denoising, and (3) a sequential combination of k-means followed by BM3D. Figure 6 illustrates the effect of these methods on two representative synthesized images from the ITB-mBatik dataset, along with their corresponding original images. The figure is organized into five columns: (1) the raw synthesized image before denoising, (2) the result after k-means denoising, (3) after BM3D denoising, (4) after sequential k-means and BM3D denoising, and (5) the original image. For this high-quality dataset, the denoising process improves visual clarity and structural consistency of the generated batik patterns, resulting in fewer dot-like patterns in the images, while the sequential combination of both methods yields

the best results.

Figure 4. Synthesized batik patterns from the ITB-mBatik dataset.

Figure 5. Synthesized batik patterns from internet-sourced batik references

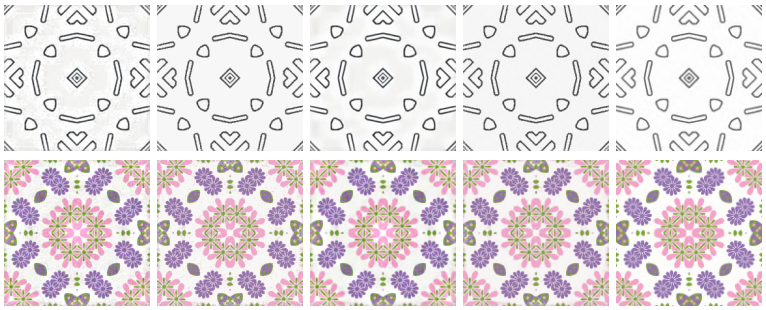


Figure 6. Denoising results for two synthesized images from the ITB-mBatik dataset. Each row shows: (1)synthesized image before denoising, (2) synthesized image after k-means denoising, (3) synthesized image after BM3D denoising, (4) synthesized image after k-means followed by BM3D denoising, and (5) the original training image.

Figure 7 shows all 30 synthesized images after applying the combined k-means and BM3D denoising method, where each image is generated from a single training image in the ITB-mBatik dataset. Compared to the results before denoising in Figure 4, this post-processing step improves image quality, particularly by reducing the noise and enhancing pattern smoothness.

Table 1 reports the Fréchet Inception Distance (FID) scores for both datasets under different denoising strategies. In this study, the FID is computed by comparing the distribution of all synthesized images with the distribution of the corresponding real images for each dataset (ITB-mBatik and internet-sourced batik). Thus, the FID is calculated at the dataset level rather than on individual images, reflecting how well the model captures the overall data distribution. The FID metric measures similarity in feature distributions, where lower values indicate better quality. A high FID score suggests a large difference between generated and real images, often due to the noise or structural inconsistencies, indicating that the model has not accurately captured the underlying patterns. In contrast, a lower FID indicates improved similarity in texture, color distribution, and overall structure.

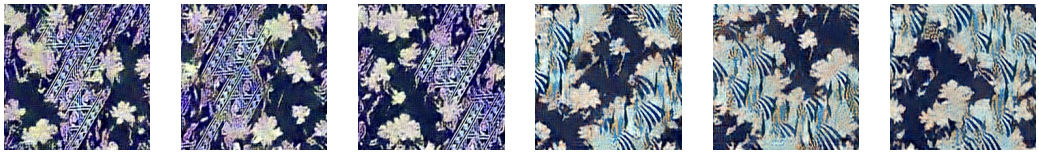
For the ITB-mBatik dataset, the FID score decreases from 158.28 to 125.33 after k-means denoising, 119.03 after BM3D, and 88.90 with the combined method, indicating a better match with the real data distribution. This suggests that denoising enhances both visual quality and statistical similarity for this dataset. In contrast, for the internet-sourced dataset, the initial FID of 74.34 does not improve consistently after denoising, increasing to 91.74 with k-means, slightly decreasing to 74.02 with BM3D, and reaching 82.91 with the combined method. This indicates that denoising may not always be beneficial, particularly when the input images already contain variability or lower pixel clarity. Therefore, only FID values are reported for this dataset without additional visual examples.

Figure 7. Synthesized batik images from the ITB-mBatik dataset after applying sequential denoising: k-means followed by BM3D.

Table 1. FID scores for synthesized images generated by the generative ConvNet from the ITB-mBatik and internet-sourced datasets. Generative ConvNet: original generative ConvNet output before denoising, K: after k-means denoising, B: after BM3D denoising, K+B: after sequential k-means and BM3D denoising.

Dataset	generative ConvNet	K	B	K+B
ITB-mBatik	158.28	125.33	119.03	88.90
Internet-sourced	74.34	91.74	74.02	82.91

The results of the Experiment 2 are presented in Figure 8, which shows blended batik patterns generated from pairs of training images. The first three images are produced using images (a) and (c) from Figure 2, while the remaining three are generated from images (b) and (c). These results demonstrate the model's ability to combine color and texture characteristics from different images into coherent and visually appealing batik motifs, particularly when the input images are carefully selected to have complementary color and texture characteristics.



(a) (b) (c) (d) (e) (f)

Figure 8. Blended batik patterns generated from two training images. (a), (b) and (c) were synthesized from training images (a) and (c) in Figure 2, while (d), (e) and (f) were synthesized from training images (b) and (c) in Figure 2.

Overall, the generative ConvNet demonstrates the ability to learn and reproduce batik patterns from both single images and image pairs. In Experiment 1, the model generates

structurally consistent patterns, while in Experiment 2, it successfully produces blended designs when the input images share complementary color and texture features. However, noise from the Langevin sampling process affects both visual quality and FID evaluation. The application of denoising, particularly the combination of k-means and BM3D, improves the results for the high-quality ITB-mBatik dataset, but shows limited or inconsistent improvement for the internet-sourced batik images.

5. Conclusions

This study demonstrates that the generative ConvNet is capable of synthesizing rich and realistic batik patterns, effectively capturing color and structural characteristics from both high-quality (ITB-mBatik) and low-quality (internet-sourced batik) datasets. In Experiment 1, the model successfully generates convincing batik patterns from a single training image, showing its ability to learn detailed texture representations. In Experiment 2, the model produces blended batik patterns by learning from pairs of images. The generative ConvNet produces visually appealing blended batik patterns, particularly when the source images share complementary color and texture; otherwise, the outputs become less coherent. To improve structural consistency, future work may incorporate a symmetry-aware loss, which is well-suited for preserving the geometric characteristics of traditional batik motifs.

The results also show that denoising plays an important role in improving the quality of synthesized images. The combination of k-means and BM3D significantly enhances image quality for the ITB-mBatik dataset, reducing the FID score from 158.29 to 88.90. However, this improvement is not observed for low quality internet-sourced batik dataset, indicating that the effectiveness of denoising depends on the quality and consistency of the input data. Further improvements may be achieved by exploring more advanced approaches, such as denoising diffusion probabilistic models [8], or classical techniques including non-local means and wavelet thresholding, as well as improved model design, to better generate batik patterns while preserving key visual characteristics.

References

[1] R. Azhar, D. Tuwohingide, D. Kamudi, N. Suciati, *et al.*, "Batik image classification using SIFT feature extraction, bag of features and support vector machine," *Procedia Computer Science*, vol. 72, pp. 24–30, 2015.

[2] H. Situngkir, "The computational generative patterns in indonesian batik," *Available at SSRN 1137166*, 2008.

[3] Chrystian and Wahyono, "Sp-batikgan: An efficient generative adversarial network for symmetric pattern generation.," *CoRR*, 2023.

[4] O. Octadion, N. Yudistira, and D. Kurnianingtyas, "Synthesis of batik motifs using a diffusion-generative adversarial network," *Multimedia Tools and Applications*, pp. 1–32, 2025.

[5] 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)*, pp. 63–67, IEEE, 2019.

[6] P. Ardhianto, Y. P. Santosa, and G. Anandhita, "Enhancing the traditional batik design practices: An approach to batik motif design using artificial intelligence.," *IAENG International Journal of Computer Science*, vol. 52, no. 2, 2025.

[7] S. I. Ishak, T. Haryanto, T. Widodo, and A. B. Santoso, "Explorasi pola batik baru dengan deep convolutional algorithme generative adversarial networks (dcgans)," *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, vol. 18, no. 1, pp. 40–44, 2023.

[8] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[9] J. Xie, Y. Lu, S.-C. Zhu, and Y. Wu, "A theory of generative convnet," in *International Conference on Machine Learning*, pp. 2635–2644, PMLR, 2016.

[10] S. C. Zhu, Y. Wu, and D. Mumford, "Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling," *International Journal of Computer Vision*, vol. 27, pp. 107–126, 1998.

[11] Y. Lu, S.-C. Zhu, and Y. Wu, "Learning frame models using cnn filters," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[12] Y. N. Wu, J. Xie, Y. Lu, and S.-C. Zhu, "Sparse and deep generalizations of the frame model," *Annals of Mathematical Sciences and Applications*, vol. 3, no. 1, pp. 211–254, 2018.

[13] Z. K. Huang, S. Q. Wang, and L. Y. Hou, "Segmentation of color textures using k-means cluster based wavelet image fusion," *Applied Mechanics and Materials*, vol. 20, pp. 209–214, 2010.

[14] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[15] C. Chrystian, "ITB-mBatik Dataset," 2023.

[16] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 689–692, 2015.