

Design and Development of an Acehese Batik Motif Classification System Based on Convolutional Neural Network and Explainable AI

Reyhan Putra Syahmi¹, M. Khadafi², Azhar³, Syarifah Annisa Salsabila⁴

^{1,2,3} Information Technology and Computer Departement, Politeknik Negeri Lhokseumawe, Indonesia

⁴ Department of Fashion Design, Faculty of Engineering, State University of Medan, Indonesia

*Corresponding Author: mkhadafi@pnl.ac.id

Article info: Received 10/08/2026, Revised 24/08/2026, Accepted 19/10/2026

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Abstract

Batik Aceh is an Indonesian intangible cultural entity characterized by high visual morphological complexity and historical depth. Conventional motif identification processes face limitations due to observer subjectivity, high spectral variance, and the intricate geometric structures composing the ornaments. Therefore, this study aims to design an automated digital image classification architecture utilizing the Convolutional Neural Network (CNN) algorithm via VGG16 implementation empowered by Transfer Learning based on ImageNet pre-trained weights. To demystify the deep learning black-box nature, the computational ecosystem is integrated with Explainable AI using Gradient-weighted Class Activation Mapping (Grad-CAM) for visual inference transparency, alongside an interactive Virtual Try-On module deployed on the Streamlit web platform. The classification scope encompasses four distinct classes: Bukan Batik (Non-Batik), Pinto Aceh, Pucok Reubong, and Rencong. The dataset comprises 700 balanced images (175 per class) sourced from the Roboflow public repository and partitioned into 80% training (560 images), 10% validation (68 images), and 10% testing (72 images). The model was compiled with the Adam optimizer (learning rate 10^{-4}), Categorical Cross-Entropy loss, and supervised by dynamic callbacks (EarlyStopping and ReduceLROnPlateau). Empirical evaluation on 72 unseen test images yielded an overall accuracy of 97.00%, an average precision of 97.50%, a recall of 97.00%, and an F1-score of 97.00%. Specifically, the Pinto Aceh and Pucok Reubong motifs achieved a perfect 100% score across all metrics, while Rencong reached a recall of 88.89%. Comprehensive Black-Box testing across 10 functional interface scenarios and White-Box code logic verification demonstrated 100% operational reliability, confirming that the proposed system is highly robust for digital cultural heritage preservation.

Keywords: : Batik Aceh, Computer Vision, Convolutional Neural Network, VGG16, Transfer Learning, Streamlit, Explainable AI

1. Introduction

Batik is an Indonesian intangible cultural heritage that has been officially recognized by UNESCO since 2009. One manifestation of Nusantara textile art that possesses high aesthetic value, distinctive ornamentation, and deep philosophical depth is Acehese Batik [1]. Fundamentally different from the batik patterns of Java, which are heavily influenced by classical figurative traditions, Acehese Batik originates from Islamic culture, which upholds the principle of aniconism — the prohibition of directly depicting living beings visually. Consequently, the ornamentation of Acehese Batik is dominated by a combination of abstract geometric forms, floral tendrils, and stylizations of historical Acehese cultural artifacts [2]. The most prominent and meaning-laden motifs include the Pinto Aceh motif,

which symbolizes humility and steadfastness; the Pucok Reubong motif, which represents generational continuity and hope for the future; and the Rencong motif, which serves as a symbol of the heroism, patriotism, and historical courage of the Acehese people.

Despite its high cultural value, the process of identifying Acehese Batik patterns and motifs among craftsmen, traders, and the general public still relies heavily on heuristic manual observation. Such conventional observation is vulnerable to observer subjectivity, spectral variance in textile dye coloring, and perspective distortion from fabric folds [3]. Moreover, laypeople and tourists often find it difficult to distinguish the details of distinctive local motif patterns from generic non-batik textiles [4]. Therefore, an innovation in computer vision technology is needed that can recognize Acehese Batik motifs objectively, accurately, automatically, and with public accessibility [5].

A number of previous studies have examined batik motif recognition using intelligent computing approaches. Azmi et al. developed a batik classification system based on a Convolutional Neural Network (CNN) trained from scratch on West Sumatran Tanah Liat batik, but experienced severe overfitting, in which training accuracy of 98.75% plummeted to 62.50% on test data [6]. The problem of limited local datasets was successfully mitigated through transfer learning techniques. Adithama et al. demonstrated the stability of an ImageNet pre-trained VGG16 model on batik classification, achieving an accuracy of 92.00% [7]. Alya et al. applied VGG16 fine-tuning with a small learning rate (10^{-4}) and obtained an accuracy of 89.00% [8], while Ardyani and Sari recorded an accuracy of 98.72% on Demak batik using an interactive web-based approach [9]. Furthermore, Prayoga et al. investigated CNN architecture variations for Yogyakarta batik motif classification, achieving an accuracy of 89.44% [10]. Other relevant visual recognition benchmarks using CNN on various traditional textiles include Indonesian batik pattern classification by Wona et al. [11], Batik and Sasirangan fabric identification by Maulida [12], fine-tuned multi-class CNN classification by Shanni et al. [13], and multi-class batik motif categorization by Arifin and Nurfaizah [14]. In the specific context of Acehese Batik, Utaminingsih and Sahputra applied the EfficientNet architecture with an accuracy of 98.00%, but deliberately excluded the Pinto Aceh and Rencong motifs, deeming them visually difficult to discriminate [15].

A common weakness of deep Convolutional Neural Network (deep CNN) models is their black-box mathematical character, which prevents users from verifying the logical reasoning behind the system's classification decisions [16]. To ensure prediction accountability, Explainable AI (XAI) methods such as Gradient-weighted Class Activation Mapping (Grad-CAM) are crucial to integrate in order to visualize the model's spatial attention [17]. Furthermore, system reliability in real-world recognition demands the presence of a negative class (Non-Batik) based on open-set recognition principles [18], so that the model does not force a motif classification onto non-batik images, especially under small and domain-specific dataset regimes [19].

Based on this research gap, this study designs and builds a digital Acehese Batik motif classification system based on the VGG16 architecture, utilizing ImageNet-weighted Transfer Learning techniques. This study covers a comprehensive four-class space, namely Non-Batik, Pinto Aceh, Pucok Reubong, and Rencong. The system is integrated with an Explainable AI (Grad-CAM) feature for visual interpretive transparency of the model, as well as an interactive Virtual Try-On module on the Streamlit web platform. Model performance is evaluated quantitatively using a Confusion Matrix, black-box functional testing, and white-box code logic path tracing.

2. Methods

Explaining the research chronologically, including the research design, research procedures (in the form of algorithms, Pseudocode, or other), how to test, and data acquisition [5] – [7]. The description of the course of research should be supported references, so the explanation can be accepted scientifically [2], [4]. Figures 1-2 and Table 1 are presented center, as shown below and cited in the manuscript [5], [8] – [13]. The effects of electrical discharges to acidity of HVNE and NELV has been illustrated in Figure 2(a) and the effects of breakdown voltage of NE and NELV has been illustrated in Figure 2(b).

2.1 Research Flow

This study adopts a software engineering and quantitative experimental-based intelligent computing methodology. The research flow is structured into eight main stages (Figure 1): literature study, dataset collection and curation, image preprocessing and augmentation, construction of the VGG16 architecture based on Transfer Learning, model training, model evaluation, system integration on the Streamlit web interface with Grad-CAM, and final documentation.



Figure 1. Research Methodology Flow Diagram of the Acehese Batik Classification System

2.2 Dataset and Data Split

The digital image dataset was acquired in a structured manner from the public computer vision repository Roboflow, ensuring consistency in resolution and minimizing optical noise from manual photography [19]. The total sample consists of 700 images distributed evenly across four class spaces: Non-Batik (175 images), Pinto Aceh (175 images), Pucok Reubong (175 images), and Rencong (175 images). The Non-Batik class was specifically included as a control (negative) class to measure the true-negative probability when the system is confronted with non-batik environmental objects [18].

The dataset was partitioned using the standard deep learning ratio of 80%:10%:10%:

1. Training Set: 560 images (140 images per class), used to calibrate the connection weight matrices in the classifier head layers.
2. Validation Set: 68 images (17 images per class), used to monitor loss convergence and prevent overfitting during the training process.
3. Testing Set: 72 images (18 images per class), allocated as independent (unseen) data to test model generalization through Confusion Matrix calculation.

2.3 Preprocessing and Data Augmentation

To align the input dimensions with the base VGG16 architecture, all images underwent the following preprocessing and augmentation stages:

1. Spatial Normalization: Images were resized to a spatial dimension of 224×224 pixels with 3 color channels (RGB).
2. Intensity Scalar Normalization: Pixel intensity values [0, 255] were scaled to the fractional range [0, 1] through the operation $I_{\text{norm}} = I / 255.0$.
3. Stochastic Training Data Augmentation: To multiply feature diversity under the limited dataset regime, on-the-fly augmentation was applied using ImageDataGenerator, including rotation of up to 30°, width/height shift of $\pm 20\%$, shear distortion at a ratio of 0.2, zoom ratio of 0.2, horizontal flip, and brightness adjustment within the range [0.8, 1.2].

2.4 VGG16 Transfer Learning Model Architecture

The system architecture topology utilizes the VGG16 Convolutional Base, consisting of 13 convolutional layers and 5 Max Pooling layers (Figure 2). This base layer was initialized using ImageNet pre-trained weights, with all weights frozen (frozen weights, layer.trainable = False), locking 14,714,688 built-in parameters to preserve low-level universal feature representations (edges, textures), as demonstrated in transfer learning optimization studies [20], [21], and robust vision feature extraction [22].

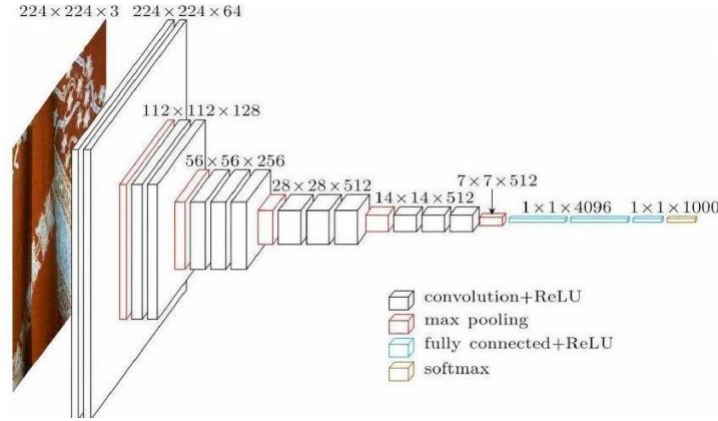


Figure 2. VGG16 Architecture Topology with Modified Classifier Head for Acehese Batik

At the end of the 5th convolutional block (block5_pool), a specially designed custom classifier head module is attached:

1. Flatten Layer: Flattens the multidimensional feature map into a 1-dimensional vector of size $7 \times 7 \times 512 = 25,088$ units.
2. Primary Dense Layer: Consists of 256 neuron units with ReLU (Rectified Linear Unit) activation to learn high-level combinatorial interactions of motif features. This layer is regularized by Dropout (ratio 0.5) to randomly disconnect 50% of neurons.
3. Secondary Dense Layer: Consists of 128 neuron units with ReLU activation, functioning as a representation bottleneck layer, equipped with Dropout (ratio 0.5).
4. Output Layer: 4 neuron units representing the 4 target classes with a Softmax activation function:

$$P(y = c \mid x) = \exp(z_c) / \sum_{i=1}^4 \exp(z_i) \quad (1)$$

where z_c is the logit value for class c ($c \in \{1, 2, 3, 4\}$).

The compilation configuration and training scenario of the model are summarized in Algorithm 1 and Table 1.

2.5 Explainable AI (Grad-CAM)

To address the black-box limitation of convolutional networks, the system applies the Gradient-weighted Class Activation Mapping (Grad-CAM) technique [16]. The algorithm computes the gradient of the target class logit score y^c with respect to the activation feature map A^k at the final convolutional layer (block5_conv3):

$$\alpha_k^c = (1/Z) \sum_i \sum_j (\partial y^c / \partial A_{i,j}^k) \quad (2)$$

where Z denotes the spatial dimension of the activation map ($i \times j$). The feature importance weight α_k^c is linearly combined and filtered through the ReLU function to preserve spatial features that positively contribute to the decision for class c :

$$L_{\text{Grad-CAM}}^c = \text{ReLU}(\sum_k \alpha_k^c A^k) \quad (3)$$

The resulting heatmap is normalized to the range $[0, 1]$, re-interpolated to the size of the original image, and overlaid (alpha blending) on top of the input image to reveal the specific motif areas referenced by the model.

Algorithm 1 Construction and Compilation of the VGG16 Transfer Learning Model

```

Require: Image tensor  $X \in \mathbb{R}^{(224 \times 224 \times 3)}$ , label  $y \in \{0, 1\}^4$  (one-hot encoded).
Ensure: Trained inference classification model  $M$ .
1: base_model  $\leftarrow$  VGG16(weights='imagenet', include_top=False, input_shape=(224,224,3))
2: for layer in base_model.layers do
3:   layer.trainable  $\leftarrow$  False    // Freeze the convolutional base weights
4: end for
5: x  $\leftarrow$  Flatten()(base_model.output)
6: x  $\leftarrow$  Dense(256, activation='relu')(x)
7: x  $\leftarrow$  Dropout(0.5)(x)

```

```

8: x ← Dense(128, activation='relu')(x)
9: x ← Dropout(0.5)(x)
10: output ← Dense(4, activation='softmax')(x)
11: M ← Model(inputs=base_model.input, outputs=output)
12: M.compile(optimizer=Adam(lr=10-4), loss='categorical_crossentropy', metrics=['accuracy'])
13: return M

```

Table 1. VGG16 Model Training Hyperparameter Configuration

Input Image Dimension	224 × 224 × 3 pixels (RGB)
Base Architecture	VGG16 (ImageNet pre-trained, frozen)
Non-Trainable Parameters	14,714,688 parameters
Trainable Parameters (Classifier Head)	6,455,940 parameters
Optimizer	Adam (learning rate = 10 ⁻⁴)
Loss Function	Categorical Cross-Entropy
Batch Size	32
Maximum Epochs	50
EarlyStopping Monitor	Patience = 10 epochs on val_loss, restore_best_weights = True
ReduceLROnPlateau Monitor	Reduction factor = 0.5, Patience = 5 epochs, min_lr = 10 ⁻⁷
Model Serialization	ModelCheckpoint (save_best_only = True)

2.6 Model Performance Evaluation Metrics

Classification performance is tested using a 4×4 Confusion Matrix that maps True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values [19]. The evaluation metrics are defined as follows:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (4)$$

$$\text{Precision} = TP / (TP + FP) \quad (5)$$

$$\text{Recall} = TP / (TP + FN) \quad (6)$$

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (7)$$

3. Result and Discussions

3.1 VGG16 Model Training Dynamics

Training of the VGG16 model with 560 training images produced a stable convergence process. Figure 3 presents the accuracy and loss curves throughout the training iterations on the training and validation data. The Validation Accuracy curve shows consistently higher performance than the Training Accuracy curve in the early epoch phase. This behavior is a positive characteristic resulting from the extreme spatial augmentation applied to the training data, which makes the learning process on the training set dynamically more challenging. The stability of the continuously decreasing Validation Loss, without divergent spikes, demonstrates the effectiveness of Dropout regularization (0.5) and the dynamic EarlyStopping monitoring mechanism in preventing overfitting symptoms [7], [8].

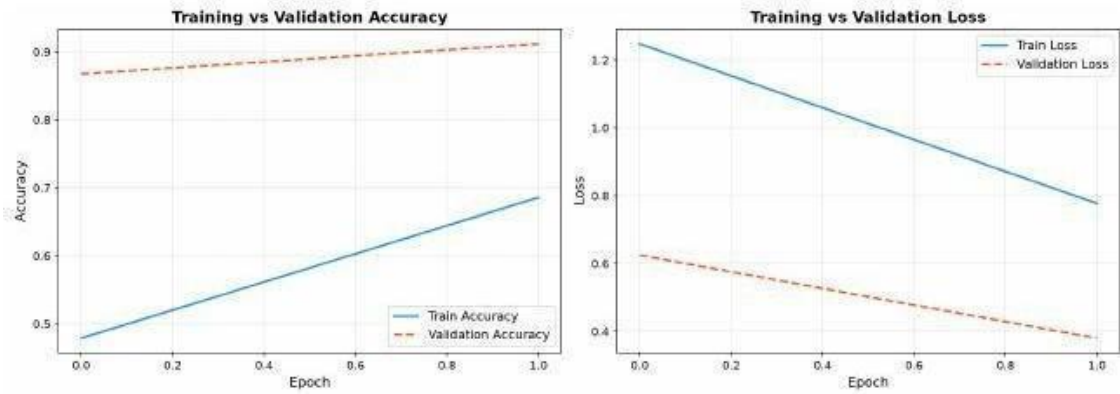


Figure 3. Accuracy and Loss Curve Graph of VGG16 Model Training

3.2 Model Evaluation Using Confusion Matrix

Model performance testing was carried out on 72 independent test images that had never been involved in the training or validation process (18 images per class). The confusion matrix of the VGG16 model's classification results is presented in Figure 4 and tabulated in Table 2.

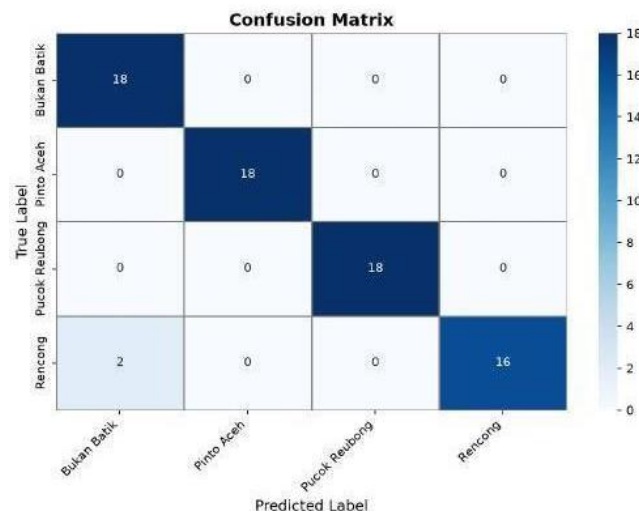


Figure 4. Confusion Matrix of Acehese Batik Motif Classification on Test Data

Table 2. Confusion Matrix of the VGG16 Model Classification Results

Actual / Predicted	Non-Batik	Pinto Aceh	Pucok Reubong	Rencong
Non-Batik	18	0	0	0
Pinto Aceh	0	18	0	0
Pucok Reubong	0	0	18	0
Rencong	2	0	0	16

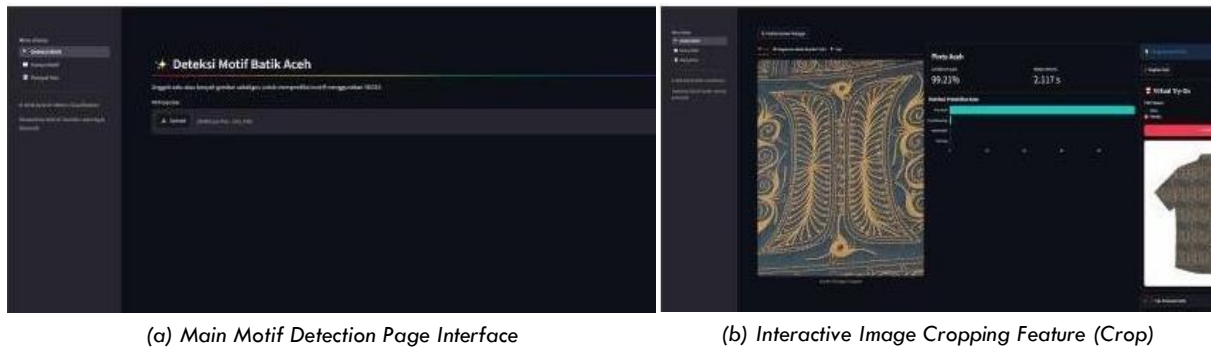
Based on Table 2, the system successfully classified 70 of the 72 test images into the correct class. The only classification deviation occurred in 2 images from the Rencong motif class, which were incorrectly predicted as Non-Batik. This is caused by the asymmetric blade-curve morphology of the Rencong motif, which lacks the regular repetition pattern found in the Pucok Reubong motif, causing certain fabric cutting orientations to be assumed as a random non-batik pattern. Detailed quantitative metrics per class are tabulated in Table 3. Overall, the VGG16 model recorded a testing accuracy of 97.00%, with a macro precision of 97.50%, recall of 97.00%, and F1-score of 97.00%. The perfect metric achievement (100.00%) for the Pinto Aceh and Pucok Reubong classes demonstrates that VGG16's convolutional feature representation is highly effective at isolating the triangular fractal symmetry characteristic of Pucok Reubong and the closed-gate curve of Pinto Aceh.

Table 3. VGG16 Model Performance Evaluation Results on Test Data per Class

Target Class	Precision (%)	Recall (%)	F1-Score (%)	Test Samples
Non-Batik	90.00	100.00	94.74	18
Pinto Aceh	100.00	100.00	100.00	18
Pucok Reubong	100.00	100.00	100.00	18
Rencong	100.00	88.89	94.12	18
Average / Total	97.50	97.00	97.00	72

3.3 Streamlit Web Interface and Grad-CAM Implementation

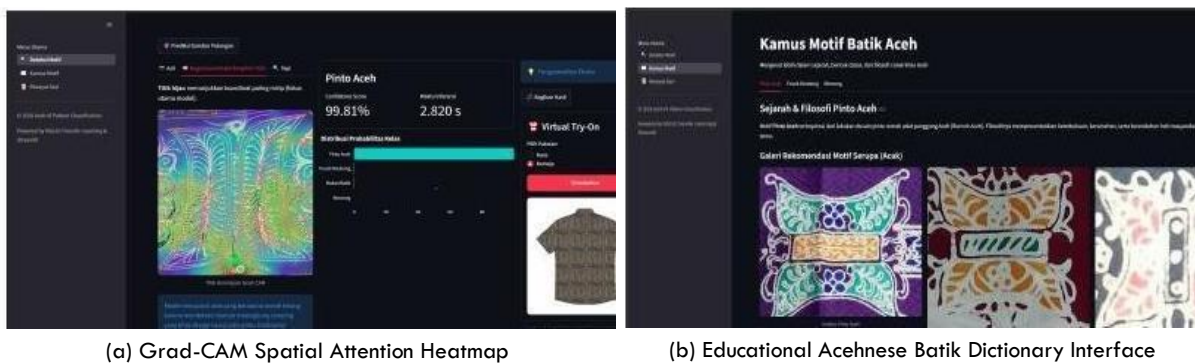
The VGG16 inference model was integrated into an interactive web interface using the Streamlit v1.45.1 framework. The interface covers image upload functionality (Figure 5a), an interactive image cropping feature to eliminate non-object backgrounds (Figure 5b), Grad-CAM Explainable AI visualization (Figure 6a), an educational motif dictionary module (Figure 6b), and session history export to a PDF document. The Grad-CAM visualization in Figure 6a shows that the model's spatial activation is strongly concentrated on the central and edge ornaments of the Acehnese Batik motif. This verifies that the model makes decisions based on the essential texture of the distinctive ornamentation, rather than on background noise or random color artifacts.



(a) Main Motif Detection Page Interface

(b) Interactive Image Cropping Feature (Crop)

Figure 5. Implementation of the Streamlit-Based Acehnese Batik Classification System Interface



(a) Grad-CAM Spatial Attention Heatmap

(b) Educational Acehnese Batik Dictionary Interface

Figure 6. Visualization of Explainable AI Features (Grad-CAM) and the Motif Dictionary Page

Note: Interface screenshots are shown in the application's original (Indonesian-language) user interface.

3.4 System Functionality Testing (Black-Box Testing)

Functional validation of the Streamlit application was carried out through 10 black-box testing scenarios adhering to standard software engineering [23] and systematic software testing methodologies [24], [25]. All scenarios were successfully executed with a 100% PASS status (Table 4).

Table 4. Web Application Functionality Testing Results Using the Black-Box Method

No	Test Scenario	System Input	Expected Result	Status
1	Access the web application URL	Open local browser	Home page loads and model is ready for inference	PASS
2	Classify a Pinto Aceh image	.jpg file of Pinto Aceh motif	Displays "Pinto Aceh" label and confidence score	PASS
3	Classify a Pucok Reubong image	.png file of Pucok Reubong motif	Displays "Pucok Reubong" label and confidence score	PASS
4	Classify a Rencong image	.jpeg file of Rencong motif	Displays "Rencong" label and confidence score	PASS
5	Classify a non-batik image	General non-batik image file	Displays "Non-Batik" label and a warning	PASS
6	Batch image processing	Upload 3 image files at once	Processes and renders results in sequence	PASS
7	Grad-CAM attention map (XAI)	Activate the XAI analysis button	Displays heatmap and visual attention justification	PASS
8	Virtual Try-On simulation	Batik image and clothing type	Renders batik texture simulation onto a clothing model	PASS
9	Motif Dictionary navigation	Click the Dictionary sidebar menu	Presents philosophical narrative of 3 Acehese Batik motifs	PASS
10	Export session history	Click the export report button	Downloads a PDF session recap report file	PASS

3.5 Code Structure Testing (White-Box Testing)

White-box testing was carried out using the Basis Path Testing technique on the main image processing function (`render_detection_page()`). Control Flow Graph analysis produced a cyclomatic complexity of $V(G) = 6$, indicating 6 fundamental independent paths (Table 5). All independent path scenarios were successfully executed without any logic errors or memory leaks, proving that the program architecture is highly robust and reliable.

Table 5. White-Box (Basis Path) Testing Results on the Image Processing Function

Path	Node Sequence	Test Input Condition	Logic Execution Result	Status
1	1 → 15	Empty input (no file uploaded)	System displays upload instructions without error	PASS
2	1 → 2 → 15	Image queue loop ends	Iteration cycle terminates completely	PASS
3	1 → 2 → 3 → 14 → 2	Corrupt / invalid file format	Exception handling active and skips the file	PASS
4	3 → 4 → 5 → 7 → 9 → 10 → 11 → 13	Crop mode active, non-batik image	Image cropped, Non-Batik class detected	PASS
5	3 → 4 → 6 → 8 → 9 → 10 → 12 → 13	Uncropped mode, valid batik image	Direct inference, Acehese Batik motif detected	PASS
6	3 → 4 → 5 → 7 → 9 → 10 → 12 → 13	Crop mode active, valid batik image	Bounding box processed, accurately classified	PASS

3.6 Discussion and Position of Findings

The 97.00% accuracy achieved in this study proves that the combination of the VGG16 architecture with the Transfer Learning paradigm is highly effective for distinguishing Acehese Batik motifs under limited dataset conditions (700 images). This achievement surpasses the previous conventional K-NN model, which experienced a performance degradation of up to 50% [1], and overcomes the overfitting obstacle encountered when training a CNN from scratch, which recorded a test accuracy drop of up to 62.50% [6]. Furthermore, this study successfully addresses the limitations of the Utaminingsih and Sahputra study [15], which excluded the Pinto Aceh and Rencong

motifs; in this study, both motifs were successfully classified with high precision, with Pinto Aceh even recording a perfect score of 100.00%. The presence of the Non-Batik class and the Grad-CAM transparency module provide a high level of operational assurance for practical application in digital cultural heritage preservation.

4. Conclusion

4.1 Conclusion

Based on the design, implementation, and series of tests that have been carried out, it can be concluded that the Acehese Batik motif classification system based on a Convolutional Neural Network using the VGG16 architecture and ImageNet Transfer Learning has been successfully built and integrated into the interactive Streamlit v1.45.1 web interface. Reconstruction of the classifier head layer with Dropout regularization (0.5) and the application of dynamic callback monitoring proved effective in guiding model convergence without overfitting symptoms. Quantitative evaluation on 72 independent test images produced an overall accuracy of 97.00%, an average precision of 97.50%, a recall of 97.00%, and an F1-score of 97.00%. The model achieved perfect performance (100%) for the Pinto Aceh and Pucok Reubong motifs, and demonstrated accountable spatial transparency through Grad-CAM visualization. Black-box functional testing on 10 interface scenarios and white-box logic testing were all successful, with a 100% success rate, proving that the system is highly reliable for supporting the digitalization and preservation of local Acehese textile culture.

4.2 Suggestions

For future system development and refinement, it is recommended to:

1. Expand the quantity and angle variation of the image dataset acquisition, particularly for the Rencong motif, to improve the recall value and minimize classification errors on extreme fabric orientations.
2. Add coverage of other distinctive Acehese motif classes, such as Bungong Jeumpa, Pintu Ulee Kareng, Awan Berarak, and the Gayo motif, to make the system's coverage more representative.
3. Apply advanced fine-tuning techniques by measuredly unfreezing the 5th convolutional block of VGG16 after initial convergence, to refine local textile domain feature extraction.
4. Explore comparative benchmarking with contemporary lightweight neural network architectures, such as MobileNetV3 or Vision Transformer (ViT), for computational optimization on mobile devices.

REFERENCES

- [1] N. Fajira, Indrawati, dan Aswandi, 2022, "Klasifikasi Citra Batik Aceh Menggunakan Metode K-Nearest Neighbor (K-NN) Berbasis Android," *Jurnal Teknologi Rekayasa Informasi dan Komputer (JTRIK)*, vol. 6, no. 1, pp. 45–52, Politeknik Negeri Lhokseumawe, ISSN: 2797-1724.
- [2] A. Zayyan dan D. Afriadi, 2023, "Changes and Adaptations of Acehese Pinto Motifs in Modern Craft Arts," in *Proc. 2nd Int. Conf. on Aceh Civilization (ICAC 2023)*, pp. 112–120, Banda Aceh, Indonesia.
- [3] A. R. Dani dan I. Handayani, 2024, "Klasifikasi Motif Batik Yogyakarta Menggunakan Metode GLCM dan CNN," *Jurnal Teknologi Terpadu*, vol. 10, no. 2, pp. 142–156, doi: 10.54914/jtt.v10i2.842.
- [4] D. Anastasya, B. P. Adhi, dan R. Prasetyo, 2024, "Implementasi Metode Convolutional Neural Network (CNN) Dalam Klasifikasi Motif Batik," *Nuansa Informatika*, vol. 18, no. 1, pp. 78–87.
- [5] W. N. Afifah dan V. Lusiana, 2025, "Klasifikasi Jenis Batik Semarang Menggunakan Metode Convolution Neural Network (CNN)," *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika (JIPI)*, vol. 10, no. 1, pp. 542–553, doi: 10.29100/jipi.v10i1.5901.
- [6] K. Azmi, S. Defit, dan Sumijan, 2023, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," *Jurnal Unitek*, vol. 16, no. 1, pp. 25–36, doi: 10.52072/unitek.v16i1.512.
- [7] A. Adithama, B. Y. Dwiandiyanta, dan Wiadji, 2023, "Klasifikasi Batik Menggunakan Pre-trained VGG16 dengan Transfer Learning," *Jurnal Teknologi Informasi*, vol. 7, no. 2, pp. 120–129.
- [8] R. F. Alya, M. Wibowo, dan Paradise, 2023, "Classification of Batik Motif Using Transfer Learning on Convolutional Neural Network (CNN)," *Jurnal Teknik Informatika (JUTIF)*, vol. 4, no. 1, pp. 161–170, doi:

10.20884/1.jutif.2023.4.1.728.

- [9] S. S. F. Ardyani dan C. A. Sari, 2024, "A Web-Based for Demak Batik Classification Using VGG16 Convolutional Neural Network," *Applied Science, Engineering and Technology (ASSET)*, vol. 6, no. 4, pp. 312–324.
- [10] A. Prayoga, D. R. S. Saputro, and B. Prasetyo, 2023, "Arsitektur Convolutional Neural Network untuk Model Klasifikasi Citra Batik Yogyakarta," *Journal of Applied Computer Science and Technology (JACOST)*, vol. 4, no. 2, pp. 82–89, doi: 10.52158/jacost.v4i2.534.
- [11] M. M. A. Wona, A. S. Adam, dan F. A. Putri, 2023, "Klasifikasi Batik Indonesia Menggunakan Convolutional Neural Network (CNN)," *Jurnal Rekayasa Teknologi Informasi (JURTI)*, vol. 7, no. 2, pp. 185–194.
- [12] I. Maulida, 2021, "Klasifikasi Kain Khas Batik dan Kain Khas Sasirangan Menggunakan Metode Convolutional Neural Network," Skripsi, Jurusan Teknik Informatika, Universitas Muhammadiyah Banjarmasin.
- [13] F. M. Shanni, A. B. Setiawan, dan M. R. A. Pratama, 2026, "Classification of Indonesian Batik Motifs using CNN VGG16 with Transfer Learning and Fine-Tuning," *Jurnal Sistem Informasi (Sistemasi)*, vol. 15, no. 6, pp. 2228–2243, doi: 10.32520/stmsi.v15i6.3512.
- [14] S. Arifin dan Nurfaizah, 2024, "Klasifikasi Motif Batik Menggunakan Metode Convolutional Neural Network (CNN) Dengan Multi Class Classification," *Jurnal Ilmiah IT CIDA*, vol. 10, no. 1, pp. 55–66.
- [15] S. R. Utaminingsih dan I. Sahputra, 2024, "Pengenalan Otomatis Pola Batik Aceh Berbasis Arsitektur EfficientNet," *Jurnal Teknologi Informasi dan Komputer*, vol. 10, no. 3, pp. 215–224.
- [16] C. J. Anders, L. Weber, D. Neumann, W. Samek, K.-R. Müller, and S. Lapuschkin, 2026, "Software for Dataset-wide XAI: From Local Explanations to Global Insights with Zennit, CoRelAy, and ViRelAy," *PLOS ONE*, vol. 21, no. 2, p. e0298412, doi: 10.1371/journal.pone.0298412.
- [17] K. Muchtar, M. F. Nugraha, F. Arnia, and R. Muharar, 2025, "Edge AI-Based Detection for Defective Coffee Beans Using Deep Learning and Streamlit Framework," *IEEE Access*, vol. 13, pp. 14205–14218, doi: 10.1109/ACCESS.2025.3421102.
- [18] A. Bendale and T. E. Boulton, 2016, "Towards Open Set Deep Networks," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1563–1572, doi: 10.1109/CVPR.2016.173.
- [19] L. Brigato, B. Barz, L. Locchi, and J. Denzler, 2022, "Image Classification with Small Datasets: Overview and Benchmark," *IEEE Access*, vol. 10, pp. 121281–121301, doi: 10.1109/ACCESS.2022.3223128.
- [20] Murinto, S. Winiarti, dan I. Faisal, 2024, "Particle Swarm Optimization Algorithm for Hyperparameter Convolutional Neural Network and Transfer Learning VGG16 Model," *Journal of Computer Science and Information Technology (JCoSITTE)*, vol. 5, no. 1, pp. 474–480, doi: 10.37859/jcositte.v5i1.6841.
- [21] D. M. Asriny dan R. Jayadi, 2023, "Transfer Learning VGG16 for Classification Orange Fruit Images," *Journal of Soft Computing and Mathematics Systems (JSMS)*, vol. 13, no. 1, pp. 206–217.
- [22] H. Ardiansyah dan T. Desyani, 2025, "Transfer Learning Menggunakan Model VGG16 untuk Klasifikasi Citra Hewan," *Jurnal Pustaka AI*, vol. 5, no. 2, pp. 441–448.
- [23] R. S. Pressman and B. R. Maxim, 2020, *Software Engineering: A Practitioner's Approach*, 9th ed., McGraw-Hill Education, New York, NY, USA, ISBN: 978-1259872976.
- [24] G. J. Myers, C. Sandler, and T. Badgett, 2011, *The Art of Software Testing*, 3rd ed., John Wiley & Sons, Inc., Hoboken, NJ, USA, doi: 10.1002/9781119202486, ISBN: 978-1-118-03196-4.
- [25] A. Escalante-Viteri and D. Mauricio, 2025, "Artificial Intelligence in Software Testing: A Systematic Review of a Decade of Evolution and Taxonomy," *Algorithms*, vol. 18, no. 3, p. 112, doi: 10.3390/a18030112.