Ingénierie des Systèmes d'Information
Vol. 31, No. 10, October, 2026, pp. 3497-3505
Journal homepage: http://iieta.org/journals/isi

# A Feature-Fusion Framework for Batik Motif Classification Using a Deep Convolutional Autoencoder, Hu Moments, and an RBF-SVM

Muhammad Faqih Dzulqarnain[1,2*], Abdul Fadlil[1], Imam Riadi[1]

[1] Doctoral Program in Informatics, Universitas Ahmad Dahlan, Yogyakarta 55191, Indonesia
[2] Department of Information Technology, Politeknik Aisyiyah Pontianak, Pontianak 78124, Indonesia

Corresponding Author Email: 2437083007@webmail.uad.ac.id

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/isi.311021

**Received:** 21 February 2026
**Revised:** 18 September 2026
**Accepted:** 30 September 2026
**Available online:** 31 October 2026

***Keywords:***
*Batik motif classification, deep convolutional autoencoder, Hu moment invariants, feature fusion, support vector machine, cultural heritage image analysis*

**ABSTRACT**

Batik motif classification is relevant to the digital documentation of Indonesian cultural heritage, but visual variation in motif orientation, scale, and detail complicates automated recognition. This study examines a feature-fusion pipeline that combines a learned image representation with a handcrafted shape descriptor for classifying four Batik motifs. The experiments use a public corpus of 680 images, comprising 170 images each of Corak Insang, Dayak, Ikat Celup, and Megamendung. Images are resized to 224 × 224 pixels and normalized. A deep convolutional autoencoder (DCAE) produces a 128-dimensional latent representation, while the seven Hu moment invariants provide a complementary shape descriptor. These features are concatenated into a 135-dimensional vector and classified using a support vector machine (SVM) with a radial basis function (RBF) kernel. The SVM parameters are selected through a grid search over 12 combinations of the penalty parameter and kernel width. Evaluation includes a random 80:20 partition, five-fold and ten-fold cross-validation, and repeated random sampling at training proportions of 80%, 70%, and 60%. The proposed pipeline achieves 100% accuracy on the reported 80:20 partition; results across the evaluation schemes range from 89.3% to 100%. Ablation results also show progressively higher accuracy when the feature components are combined. However, an integrity audit identifies exact and near-duplicate images across the training and test partitions. The reported scores therefore characterize performance on this corpus and do not establish generalization to independently collected Batik images.

## 1. INTRODUCTION

Image classification is one of the main challenges in the field of computer vision, which aims to identify and categorize objects in digital images [1]. Advancements in this technology have opened many opportunities in various sectors, such as Javanese Batik [2, 3] and industry [4]. One of the interesting applications is in the recognition of patterns on cultural objects, especially Kalimantan's Batik [5]. Batik, as Indonesia's cultural heritage, has a wealth of patterns and very diverse details [6, 7]. The classification of Batik patterns requires an effective and efficient approach to recognize the various variations of shapes and patterns that exist in each type of Batik.

Batik patterns have a high complexity, as each motif can vary depending on the technique, material, and region of manufacture [8]. This diversity of patterns makes the task of classifying Batik images a challenge in computer vision. Although various methods based on convolutional neural networks (CNNs) have been used to identify Batik patterns, there are still obstacles in reducing the loss value in the training process [9, 10]. The high loss value can affect the model's ability to recognize Batik patterns accurately, so further optimization in the training process is needed.

Several new approaches have been developed to address this problem, including the augmentation techniques to increase detail of selection feature and the application of trained CNN models [11]. In addition, feature extraction techniques such as deep convolutional autoencoder (DCAE) [12-14] and Hu moments [9, 15, 16] have also begun to be used to improve classification efficiency. DCAE functions to perform unsupervised feature learning by producing a denser and more discriminative image representation, as well as reducing noise in the image [17]. This technique has been shown to be effective in improving the quality of image representation for classification. On the other hand, Hu moments are used to capture shape characteristics that are invariant to geometric changes such as rotation, scaling, and translation, which is crucial in handling variations in Batik imagery that have different patterns of position or orientation [18, 19].

DCAE is a neural network designed to extract image features unsupervised [14]. DCAE consists of two main parts, which are the encoder and the decoder [20]. Encoders compress the input image into a representation of features that are lower in dimensions, while the decoder reconstructs the image from that representation [20]. This process allows the model to learn important characteristics of imagery without the need for labels, which is especially useful when datasets are not available in large quantities or are difficult to label [21].

In the context of Batik image classification, DCAE can reduce noise and produce denser features, so that it can improve the model's ability to distinguish various Batik patterns more accurately. The advantage of DCAE is its ability to capture hidden information in imagery and produce better representations in the face of various geometric variations.

Hu moments is a shape descriptor used to measure the geometric characteristics of an image in a way that is not affected by rotation, scaling, or translation [18, 22]. Hu moments refers to a group of low-order moments that are calculated from imagery to capture important shape features [23]. Using Hu Moments, the classification system can handle variations in Batik pattern shapes that appear in different positions and sizes. One of the main advantages of Hu moments is its ability to produce representations of shapes that are invariant to the geometric transformations often found in Batik images. For example, when a Batik pattern is rotated or resized, Hu moments can still recognize the shape accurately, making it a very effective tool for pattern recognition. This technique allows the model to be more robust in dealing with the variations in shape that occur in the Batik image.

The process of extracting image features will be given the development of a combination between DCAE and Hu moments provides new opportunities that have several advantages, DCAE focuses on learning features in depth, capturing texture and spectral information from Batik images, while Hu moments ensures that the characteristics of the main shape are maintained despite geometric transformations. The integration of these two techniques can result in a better and more comprehensive representation of features, which is crucial for accurate classification of Batik imagery. DCAE will handle noise and improve the quality of image representation, while Hu moments will capture invariant shapes that provide resistance to geometric variations in Batik patterns.

Hybrid CNN-SVM approaches have been used for image classification [24-26]. One such study evaluated the fusion of handcrafted features and CNN-derived representations for SVM-based classification [25]. CNNs function to extract features in depth, while SVM is used as a stable and efficient classification model [27-30]. The classifier is tuned by an explicit grid search over $C$ and $\gamma$ rather than adopted with default settings, and both the search space and the selected configuration are reported. By combining these two methods, classification performance can be significantly and steadily improved. This hybrid technique helps to reduce the value of losses that occur during training, so that the resulting model becomes more accurate and robust in handling complex Batik images.

This research seeks to create a hybrid model combining CNN-SVM with the integration of DCAE and Hu moments, aiming to enhance the classification accuracy of Batik images and reduce errors. By adopting this method, it is anticipated that improved results will be achieved compared to previous Batik image classification techniques, while also contributing new insights to the fields of computer vision and pattern recognition within image datasets.

The contribution of this study is threefold. First, it specifies a compact fused descriptor of 135 dimensions, combining a learned representation with an analytic shape descriptor, and reports the architecture in full. Second, it selects the classifier configuration by an explicit grid search whose space and outcome are both reported, rather than adopting default settings. Third, it evaluates the pipeline under six partitioning schemes and examines, in the concluding section, how the composition of the corpus itself bounds the interpretation of the resulting figures.

Deployment of such a classifier in a mobile application would additionally raise questions of data protection and forensic readiness, including interception during transmission, malware targeting mobile banking components, and the availability of forensic tooling for incident analysis [31-34]. These concerns lie outside the scope of the present study but bear on any field deployment of the pipeline described here.

## 2. RELATED WORK

In the context of Batik classification, recent research delves into the use of a DCAE [35, 36] to improve Batik classification accuracy [12, 37]. By incorporating an autoencoder into the CNN architecture, the study demonstrates enhanced feature extraction, leading to a significant improvement in classification accuracy. The authors show that DCAE models can achieve a high classification accuracy of 99.5% [12], outperforming traditional CNN approaches. The research emphasizes the power of DCAE in augmenting image data, providing sharper and more accurate features for classification tasks.

Other recent studies have explored various methods for image recognition, with a focus on improving feature extraction techniques. The first article evaluates two methods for Batik motif classification: convolutional autoencoders (CAE) and shape-based Hu moments [9, 15]. The research highlights the trade-offs between these techniques, with CAEs offering robust learning of complex latent features, while Hu moments provide invariant shape descriptors that are more interpretable. The authors emphasize the complementary strengths of these methods in capturing Batik's intricate patterns and details, suggesting that a hybrid approach combining both could optimize classification results.

Taken together, these studies show that learned and handcrafted descriptors are each effective for Batik recognition, and that combinations of the two have been examined before. The present work claims no priority for the combination itself. Its contribution lies in three properties rarely reported together: a compact fused descriptor of 135 dimensions, against the several hundred to several thousand typical of pre-trained embeddings; a representation learned from the corpus itself rather than transferred, so no external image collection enters the pipeline undocumented; and a classifier tuned by an explicit search whose space and outcome are both reported. What remains open, and what the concluding section addresses, is how the evaluation protocol itself shapes the figures obtained from small heritage corpora.

## 3. METHODOLOGY

This research proposes a hybrid approach in the classification of Batik images using a combination of DCAE and Hu moments for feature extraction, followed by model training using a hybrid CNN and SVM. The stages in this research include several main steps, ranging from loading Batik image datasets, data preprocessing, feature extraction using DCAE and Hu moments, to the image classification process using the CNN-SVM model. A detailed description of the flowchart is shown in Figure 1.

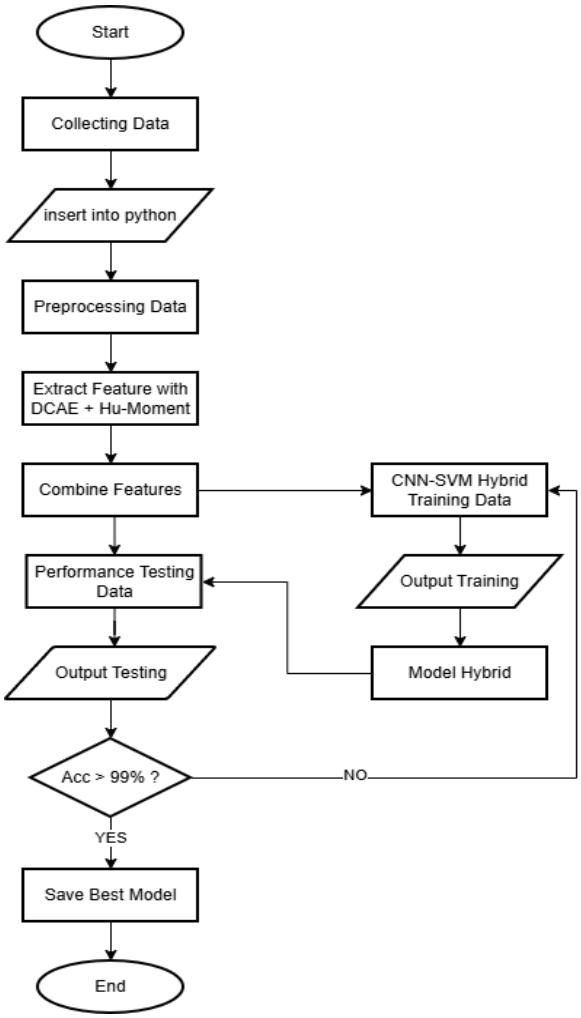


**Figure 1.** Research diagram

### 3.1 Dataset

The corpus used in this study is a publicly available collection of 680 Batik images [3, 9, 12] distributed with earlier work on the same material and reused here without modification. Reusing an existing public corpus is deliberate: it allows the pipeline proposed here to be positioned against previously reported results on identical material, which would not be possible with a newly collected private set. The collection contains four motif classes. Corak Insang, Dayak, and Ikat Celup are the three motifs produced in the largest volume in West Kalimantan, and they therefore represent the motifs a regional recognition system would encounter most often. Megamendung originates from Cirebon in West Java and is retained as an out-of-region class, giving the classifier a visually distinct reference against which the three regional motifs must be separated, as shown in Table 1.

**Table 1.** Total variation of the image dataset

| Batik Pattern | Size (Pixel) | Total Images |
|---|---|---|
| Corak Insang | 224 × 224 | 170 |
| Dayak | 224 × 224 | 170 |
| Ikat Celup | 224 × 224 | 170 |
| Megamendung | 224 × 224 | 170 |

The scope of the study follows from this composition. The objective is methodological: it examines how a learned representation and a handcrafted shape descriptor combine on a fixed corpus rather than to characterize the diversity of Indonesian Batik as a whole, which would require a corpus spanning many more provinces and motif families than the four classes available here.

The distribution of datasets was carried out randomly using a proportion of 80% for data training and 20% for data testing. This distribution aims to train models using large enough data and test model performance using data that has never been seen before.

### 3.2 Preprocessing

In the context of image pixel classification, data preprocessing plays an important role in optimizing model accuracy and ensuring robustness, especially in handling complex patterns of Batik image motifs. The preprocessing pipeline for this research is designed to address unique challenges in the introduction of Batik patterns, including texture variation, color distribution, and symbolic motifs. This methodology combines the latest techniques to ensure the extraction of features that retain the essence of traditional Batik patterns, while adapting to the needs of the computational model.

The preprocessing stages applied in this research include image size change and normalization to ensure consistency of input dimensions and data scale, followed by data augmentation techniques. This augmentation process includes vertical and horizontal flips and crop center 20%, which is essential to simulate the natural variations found in hand-crafted Batik patterns. These augmentations were strategically chosen to address challenges in the classification of Batik patterns, such as dealing with small geometric distortions and pattern alignment mismatches arising from different production techniques.

This preprocessing approach to data not only ensures that the model can recognize image patterns under a variety of conditions but is also in line with best practices in computer vision, which are specifically designed to preserve the unique and detailed features of cultural artifacts such as Batik imagery. Augmentation was applied to the training partition only, after the split, using horizontal flipping, vertical flipping, and a center crop retaining 80% of the image area. Each source image therefore yielded four training instances. The testing partition was left unaltered. The details of the augmentation are shown in the following Table 2.

To avoid data leakage, the model development was trained exclusively on the training dataset. The testing data were never involved during the model training phase and were only used for feature extraction during the evaluation stage [38]. This separation ensures that the test images play no part in fitting the model. Section 4.3 examines the extent to which the corpus itself limits what such a separation can guarantee. The training dataset, consisting of 544 images, was used for model optimization, including feature extraction through the integration of DCAE and Hu moments, and model classification using the hybrid CNN-SVM approach. The testing dataset, consisting of 136 images, was introduced only at the evaluation phase.

**Table 2.** Image preprocessing treatment details

| Split Dataset | Percentage (%) | Total Images | After Augmentation |
|---|---|---|---|
| Training | 80 | 544 | 2,176 |
| Testing | 20 | 136 | 136 |

### 3.3 Model development

The pipeline assigns each component a distinct and complementary role, as shown in Figure 2. The convolutional stack acts as the feature generator, mapping raw pixels to a hierarchy of local patterns without requiring those patterns to be specified in advance. The DCAE acts as the representation stage: by reconstructing its own input through a narrow bottleneck, it compresses those patterns into a 128-dimensional code that retains what is needed to regenerate the image and discards what is not. The Hu moments act as an explicit shape descriptor, computed analytically from the image moments and invariant to rotation and scale, which matters for Batik because a motif carries the same identity whichever way the fabric is oriented. The two sources are concatenated into a 135-dimensional descriptor, of which 128 dimensions are learned and 7 are analytic, and an SVM with a radial basis function (RBF) kernel draws the decision boundary in that fused space. The division of labor is deliberate: the analytic component supplies an invariance that a small training corpus is unlikely to teach on its own, and the classifier is kept simple so that the discriminative work rests on the descriptor rather than on a deep decision head.

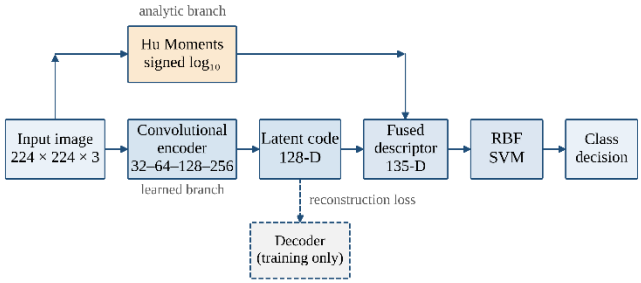


**Figure 2.** Model development design

Figure 2 shows an architecture of the proposed pipeline. Convolutional layers generate features from the input image; the autoencoder bottleneck compresses them into a 128-dimensional latent representation; seven Hu moments are computed analytically from the same image as a rotation- and scale-invariant shape descriptor; the two are concatenated into a 135-dimensional descriptor that forms the input to the RBF SVM, which produces the class decision.

DCAE performs unsupervised feature learning, producing a denser and more discriminative representation while suppressing noise, without relying on labels [39]. This property is useful on datasets of limited size, where overfitting is a recurring difficulty [21]. Hu moments complement this by capturing shape characteristics that remain stable under the rotation, scaling, and translation introduced during augmentation [15].

Once the features are extracted using DCAE and Hu moments, the results are combined and used as inputs for the CNN-SVM hybrid model. CNN functions to extract features in depth through several layers of convolution, which can recognize intricate patterns in Batik imagery. SVM is then used as a classifier to distinguish image classes based on features extracted by CNN.

The classifier hyperparameters were not fixed a priori. A grid search was carried out over the penalty term cost ($C$) $\in$ {0.1, 1, 10, 100} and the RBF kernel width $\gamma \in$ {0.001, 0.01, 0.1}, giving twelve candidate configurations. Each configuration was scored by cross-validation restricted to the training partition, so that the held-out data played no part in the selection. The configuration achieving the highest mean validation accuracy was retained and applied unchanged to the held-out data. This procedure is what the term optimized in the title refers to: the architecture is fixed by design, while the decision boundary of the final classifier is tuned by search rather than by assumption.

### 3.4 Evaluation

Two distinct losses arise in this pipeline and are reported separately, since conflating them would make either figure uninterpretable. The autoencoder is trained by minimizing the mean squared error between an input image and its reconstruction, computed over pixel values scaled to the unit interval; this quantity measures reconstruction fidelity, is dimensionless, and is not a percentage. The classification stage is not trained by that criterion: the SVM maximizes the margin under a hinge objective, and its performance is reported through accuracy, precision, recall, and F1-score. Where a loss value accompanies a classification result in this paper, it refers to the reconstruction error of the representation stage.

Evaluation follows two schemes. The primary results are obtained from a random 80/20 partition, with 544 images for training and 136 for testing. Because a single partition of a small corpus yields an estimate whose variability is unknown, the evaluation is extended to repeated schemes: five-fold and ten-fold cross-validation, and repeated random sampling with ten repetitions at training proportions of 80, 70, and 60 percent. Two points must be stated about what this extension does and does not establish. It establishes that the reported figure is not an artifact of one fortunate partition, since the estimate remains stable across partition schemes and training proportions. It does not establish generalization to unseen material, because augmentation was applied before partitioning, so variants derived from a single source image may fall on both sides of any of these partitions. Section 5 addresses this property of the corpus and its consequence for the interpretation of the reported accuracy.

Implementation details are reported so that the experiment can be repeated. The DCAE was trained outside the graphical environment, in Python 3.10 with TensorFlow 2.10.1 and Keras under Compute Unified Device Architecture (CUDA) 11.2 and CUDA Deep Neural Network library (cuDNN) 8.1.0, within a dedicated Conda environment; image operations used Open Source Computer Vision Library (OpenCV) and the supporting analysis used scikit-learn, pandas, and Matplotlib. The resulting 128-dimensional latent codes were exported and combined with the seven log-transformed Hu moments to form the 135-dimensional descriptor. Classification, hyperparameter search, and all six evaluation schemes reported in Section 4 were executed in Orange Data Mining 3.40. Preprocessing and augmentation were carried out on the Roboflow platform. All experiments were run on a workstation with an AMD Ryzen 7 5700G processor, 32 GB of system memory, and an NVIDIA GeForce GTX 1660 SUPER graphics card with 6 GB of memory.

## 4. RESULTS AND DISCUSSION

### 4.1 Dataset results

The dataset was organized on the Roboflow platform, which

records the preparation steps applied to all 680 Batik images. Figure 3 shows the resulting configuration: an 80/20 partition giving 544 training and 136 testing images, together with the preprocessing and augmentation settings.

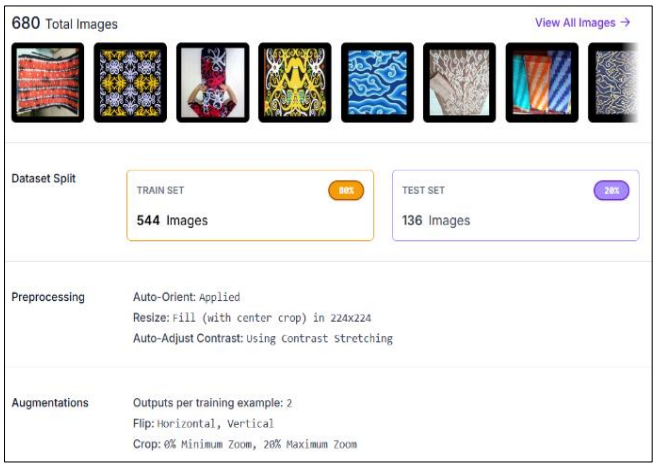


**Figure 3.** Dataset results from Roboflow

Preprocessing standardized every image in the corpus: orientation was normalized to correct variations in capture angle, images were resized to 224 × 224 pixels using a fill strategy with center cropping, and contrast stretching was applied to emphasize motif detail. These operations preserve structural and texture information while suppressing variation that carries no motif information.

Augmentation was applied exclusively to the training set, through horizontal and vertical flipping and center cropping at a ratio of 20%, simulating the changes in orientation, scale, and partial motif visibility found in handcrafted textiles. Each source image yielded four training instances, expanding the training set from 544 to 2,176 images. The test set was left unaltered.

### 4.2 Model development results

Feature extraction is completed before this stage: the latent code produced by the autoencoder, trained as described in Section 3.4, is combined with the Hu moments computed from the same image and stored as a single table. The graphical environment therefore handles classification and evaluation only, taking that table as its input. The workflow is illustrated in Figure 4.

The workflow reads the stored descriptors, passes them to the fully connected stage and the SVM, and scores the result through test and score, whose predictions are summarized in a confusion matrix. The training parameters of both stages are listed in Table 3.

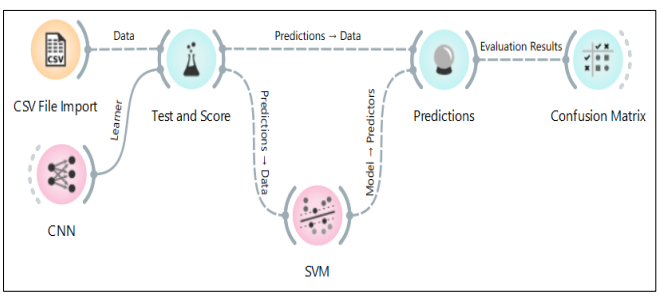


**Figure 4.** Result model from graphical data mining

**Table 3.** Details of the training layers for CNN-SVM

| Parameters | Value |
|---|---|
| Input image size | 224 × 224 × 3 |
| Pixel scaling | [0, 1] |
| Training augmentation | H-flip, V-flip, 20% center crop |
| Block 1: Conv 3 × 3, stride 2, BatchNorm, ReLU | 32 filters, 112 × 112 × 32 |
| Block 2: Conv 3 × 3, stride 2, BatchNorm, ReLU | 64 filters, 56 × 56 × 64 |
| Block 3: Conv 3 × 3, stride 2, BatchNorm, ReLU | 128 filters, 28 × 28 × 128 |
| Block 4: Conv 3 × 3, stride 2, BatchNorm, ReLU | 256 filters, 14 × 14 × 256 |
| Projection: Conv 1 × 1, then flatten | 8 filters, 1568 |
| Latent code: Dense, linear activation | 128 |
| Expansion: Dense, then reshape | 1568 → 14 × 14 × 8 |
| Blocks 1–4: Conv2DTranspose 3 × 3, stride 2, BatchNorm, ReLU | 256, 128, 64, 32 filters |
| Reconstruction: Conv 3 × 3, sigmoid | 224 × 224 × 3 |
| Skip connections | None |
| Reconstruction loss | MSE on [0, 1], dimensionless |
| Optimizer, learning rate | Adam, $1 \times 10^{-3}$ |
| Shape descriptor: Hu moments, signed $\log_{10}$ | 7 |
| Fusion rule: concatenation of latent code and moments | 128 + 7 = 135 |
| Standardization, fitted on training data only | 135 |
| Neurons in hidden layer | 100 |
| Activation | ReLU |
| Solver/optimizer | Adam |
| Regularization (L2 penalty $\alpha$) | 0.1 |
| Iterations | 400 |
| Kernel | RBF |
| SVM cost ($C$) | 0.1 |
| SVM kernel width ($\gamma$) | 0.01 |
| Numerical tolerance | 0.001 |
| SVM iteration limit | 100 |
| Output | 4 classes |

Note: CNN = convolutional neural network; DCAE = deep convolutional autoencoder; SVM = support vector machine; RBF = radial basis function; MSE = Mean Squared Error; ReLU = Rectified Linear Unit; BatchNorm = Batch Normalization; Conv = Convolution; Conv2DTranspose = Transposed Two-Dimensional Convolution; L2 = L2 regularization; Adam = Adaptive Moment Estimation.

Table 3 traces the pipeline from the input image to the class decision. Number 1 to 3 shows the input geometry and the augmentation applied to the training partition. Rows 4 to 9 describe the encoder: 4 strided convolutional blocks reduce the spatial resolution from 224 × 224 to 14 × 14 while widening the channel dimension, after which a 1 × 1 projection compresses the channels and a dense layer produces the 128-dimensional latent code. Rows 10 to 15 describe the decoder, which mirrors the encoder and reconstructs the input at full resolution. Two properties of this design determine whether the latent code carries information and are therefore stated explicitly. No skip connection joins the encoder to the decoder, so every reconstruction path is forced through the bottleneck; were such a connection present, the decoder could bypass the code, and the code would be free to carry nothing. The reconstruction is scored by mean squared error over pixel values in the unit interval, which is a dimensionless quantity and not a percentage. Rows 16 to 18 specify the fusion: seven Hu moments, transformed by a signed base-ten logarithm because the raw moments span several orders of magnitude, are concatenated with the latent code to form a 135-

dimensional descriptor, of which 128 dimensions are learned, and 7 are analytic. Standardization is fitted on the training partition alone and applied unchanged to the held-out data. Number 19 to 29 show the classification stage, in which a fully connected layer of 100 units precedes an SVM with an RBF kernel whose cost and kernel width are determined by the grid search.

With the architecture fixed as described above, the pipeline was evaluated under the schemes set out in Section 3.4. Reporting several schemes rather than one serves a specific purpose on a corpus of this size: a single partition yields a point estimate whose variability cannot be assessed, so agreement across partitioning strategies and training proportions is what distinguishes a stable result from a fortunate split. Table 4 reports accuracy, precision, recall, and F1-score for each scheme.

**Table 4.** Evaluation results of the development model under different evaluation schemes

| Evaluation Scheme | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|
| Single random split, 80/20 | 100.00 | 100.00 | 100.00 | 100.00 |
| Cross-validation, 5 folds | 99.90 | 97.50 | 97.50 | 97.50 |
| Cross-validation, 10 folds | 99.90 | 97.50 | 97.50 | 97.50 |
| Repeated random sampling, 80% | 100.00 | 100.00 | 100.00 | 100.00 |
| Repeated random sampling, 70% | 99.90 | 97.70 | 97.70 | 97.70 |
| Repeated random sampling, 60% | 89.30 | 90.70 | 89.40 | 89.40 |

The figures in Table 4 remain within a narrow band across the first five schemes and fall sharply at a training proportion of 60%. The stability rules out a single fortunate split, since cross-validation over 5 and 10 folds and repeated sampling at 80% and 70% return comparable figures; the drop of roughly ten percentage points at 60% shows the pipeline is sensitive to the volume of training material, which is expected on a corpus of this size. What the table does not establish is generalization to unseen material, because augmentation preceded partitioning: variants of one source image may fall on both sides of any such partition, so stability across them is expected regardless of performance on independently collected images.

The classifier configuration underlying these figures was selected by search rather than assumed. Table 5 reports the mean validation accuracy for each combination of the penalty term $C$ and the RBF kernel width $\gamma$ across the twelve configurations evaluated on the training partition.

Five of the twelve configurations attained the highest validation accuracy. Ties were resolved by a rule fixed before the search rather than after inspecting the results: among configurations of equal accuracy, the one with the smallest penalty term $C$ is preferred, and among those, the one with the smallest kernel width $\gamma$. The configuration retained is therefore the most strongly regularized of the tied candidates. Inference cost was not measured in these experiments, and no claim is made about it.

**Table 5.** Results from different sets of $C$ and $\gamma$

| Value $C$ | Value $\gamma$ | Accuracy (%) |
|---|---|---|
| 0.1 | 0.001 | 99.80 |
| 0.1 | 0.01 | 100.00 |
| 0.1 | 0.1 | 100.00 |
| 1 | 0.001 | 99.80 |
| 1 | 0.01 | 99.30 |
| 1 | 0.1 | 99.30 |
| 10 | 0.001 | 100.00 |
| 10 | 0.01 | 100.00 |
| 10 | 0.1 | 100.00 |
| 100 | 0.001 | 95.00 |
| 100 | 0.01 | 95.00 |
| 100 | 0.1 | 95.00 |

### 4.3 Evaluation results

Performance evaluation is analyzed through a confusion matrix, which provides a comprehensive representation of the classification results across all Batik classes. The confusion matrix reveals all classification results from the classified test data without any errors, as shown in Figure 5. These results demonstrate the strong discriminating capabilities of the proposed hybrid CNN-SVM model when enhanced with DCAE and Hu moments features.

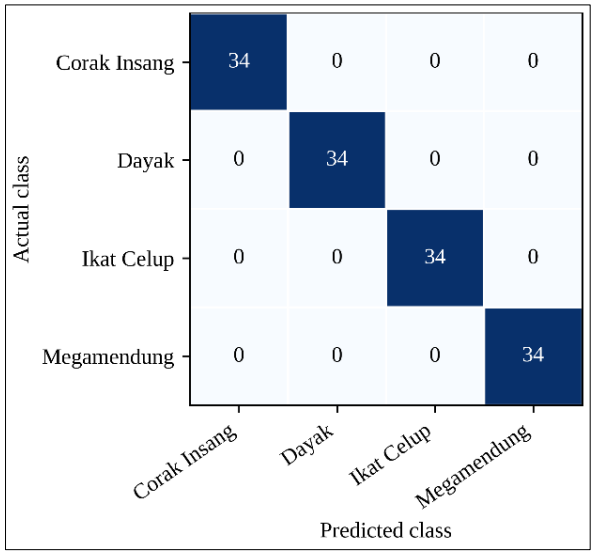


**Figure 5.** Confusion matrix of the development model

The confusion matrix contains no off-diagonal entries. Every sample of Corak Insang, Dayak, Ikat Celup, and Megamendung was assigned to its own class, giving zero false positives and zero false negatives for all four motifs.

The partitioning procedure itself was applied correctly: the test images were excluded from every stage of model fitting, including feature extraction and hyperparameter selection, and were introduced only at the evaluation stage. That procedural separation is a necessary condition for an unbiased estimate, but on this corpus, it is not a sufficient one. An integrity check reported in Section 5 shows that the corpus contains exact and near-exact duplicates distributed across both partitions, so a test image may be procedurally unseen while remaining visually identical to a training image. The perfect scores in Table 4 therefore indicate that the pipeline separates these four motif classes on this corpus, not that it would do so on independently collected material.

### 4.4 Ablation study

Table 6 isolates the contribution of each component of the descriptor. Four arms were constructed by removing one source at a time: the convolutional features alone, those features with the Hu moments, the autoencoder latent code alone, and the complete fused descriptor. All four were executed within a single evaluation on the same partition, with the SVM configuration selected in Table 5 held fixed, so that the only quantity varying between rows is the composition of the input descriptor.

**Table 6.** Ablation of the descriptor components

| Model | Accuracy (%) | F1-Score (%) |
|---|---|---|
| CNN only | 97.06 | 97.06 |
| CNN + Hu | 98.53 | 98.53 |
| CNN + DCAE | 99.26 | 99.26 |
| Full pipeline | 100.00 | 100.00 |

Note: CNN = convolutional neural network; DCAE = deep convolutional autoencoder.

The convolutional features alone misclassify four of the 136 test images. Appending the analytic moments halves that count to two. Replacing those features with the autoencoder latent code leaves a single error, which indicates that passing the representation through a reconstruction bottleneck retains information the raw features carry less compactly. The complete pipeline, in which the latent code is fused with the moments, is the only arm to classify every test image correctly. The differences are nonetheless small in absolute terms: the entire spread across the four arms corresponds to four images, so the ordering is consistent with complementarity rather than a precise measurement of each component's contribution.

Two observations bound what these results support. The confusion matrix contains no off-diagonal entries, so the data say nothing about which motifs are mutually confusable; a corpus on which every class is separated without error cannot discriminate between competing representations. The limitation is likewise the corpus rather than the pipeline: with four classes and few independent examples, a compact descriptor is an advantage precisely because it constrains the number of parameters to be fitted. Whether it retains that advantage as the number of motif classes grows is not answered here. The following subsection places the complete pipeline beside figures reported by previous studies on the same corpus.

### 4.5 Comparison results

Table 7 places the present result beside figures previously reported on the same public corpus. The comparison is informative because the underlying material is identical, and it is limited for a reason that must be stated rather than left implicit: the three comparison figures were obtained under the preprocessing and partitioning of their respective studies, which this work does not reproduce. A difference between two accuracies obtained under different evaluation procedures cannot be attributed to the methods that produced them, particularly on a corpus of this size where the partitioning strategy exerts a strong influence on the figure obtained. The rows of Table 7 should therefore be read as context for the present result rather than as a ranking of methods. The controlled comparison in this study is Table 6, in which every arm was executed under one common protocol.

**Table 7.** Comparison of research results

| Model | Accuracy (%) | Loss | Research Benchmark |
|---|---|---|---|
| Full pipeline | 100.00 | 0.0006 | This Research |
| CNN + MobileNet | 92.50 | 0.4380 | Previous Research [3] |
| CNN + Hu | 98.03 | 0.0022 | Previous Research [9] |
| CNN + DCAE | 99.00 | 0.0041 | Previous Research [12] |

Note: Loss values for the comparison studies are reproduced as reported in their respective sources; the loss function is not stated uniformly across those studies, so the column is included for completeness and is not a like-for-like comparison. CNN = convolutional neural network; DCAE = deep convolutional autoencoder

Table 7 lists testing accuracy and loss for four models evaluated on the same public corpus. The proposed pipeline reaches 100% accuracy with a reconstruction loss of 0.0006. The CNN with Hu moments reaches 98.03%, the CNN with an autoencoder 99.00%, and the CNN with MobileNet transfer learning 92.50%. As stated above, these three figures come from their own studies' protocols and are reported here as context rather than as a ranking.

The three comparison figures were each obtained under their own pre-processing and partitioning, which this study does not reproduce. A difference between accuracies measured under different procedures cannot be attributed to the methods that produced them, particularly on a corpus of this size, where the partitioning strategy strongly influences the result. Table 7 therefore serves as context for the present work. The controlled comparison is Table 6, in which every arm shared a single protocol and only the descriptor varied.

## 5. CONCLUSIONS

This study examined a hybrid pipeline in which a DCAE supplies a learned representation, seven Hu moments supply an analytic shape descriptor invariant to rotation and scale, and an SVM with an RBF kernel performs the final decision. The grid search over twelve configurations shows that the outcome is governed chiefly by the penalty term rather than by the kernel width: accuracy varies by at most half a percentage point across the three values of $\gamma$ at any fixed $C$, but falls by five percentage points when $C$ is raised to its largest value. The classifier was therefore tuned by search rather than adopted with default settings, and the configuration retained is the most strongly regularized among those attaining the highest validation accuracy. The ablation in Table 6 indicates that the two feature sources are complementary rather than interchangeable: each addition reduces the number of misclassified images, and only the fused descriptor classifies the held-out set without error.

An integrity check based on file digests and perceptual hashing indicates that the image files correspond to a considerably smaller number of independent images, the remainder being exact or near-exact duplicates, and that a large majority of the test files have a counterpart among the training files. Augmentation was moreover applied before partitioning, so variants of one source image fall on both sides of the split. Under such conditions, a classifier recognizes material it has already encountered, and an accuracy near the ceiling is the expected outcome for any competent model. The figures reported here therefore describe the behavior of the pipeline on this corpus under this partition, and are not an estimate of performance on unseen motifs.

This defines the most useful direction for further work, which concerns dataset preparation rather than model architecture. An integrity audit should precede training so that the number of independent images is known before it is reported; that number should be stated alongside the file count, since only the former reflects the available statistical power; partitioning should be performed over source images, with every variant of one image kept on the same side of the split and augmentation applied afterwards and only to the training partition; and evaluation should be repeated over several partitions and reported with its dispersion. Such a protocol will yield a lower figure than the one obtained here, and that reduction is precisely the point, since the resulting quantity is what a preservation system deployed in the field would actually deliver.

## REFERENCES


[1] Fitriani, L., Tresnawati, D., Sukriyansah, M.B. (2023). Image classification on Garutan Batik using convolutional neural network with data augmentation. JUITA: Jurnal Informatika, 11(1): 107-115. https://doi.org/10.30595/juita.v11i1.16166

[2] Fadlil, A., Riadi, I., Purwadi Putra, I.J.D.E.P. (2023). Comparison of machine learning performance using Naive Bayes and Random Forest methods to classify Batik fabric patterns. Revue d'Intelligence Artificielle, 37(2): 379-385. https://doi.org/10.18280/ria.370214

[3] Sastypratiwi, H., Muhardi, H., Yulianti. (2024). Batik recognition and classification using transfer learning and MobileNet approach. International Journal on Informatics Visualization, 8(4): 2400-2410. https://doi.org/10.62527/JOIV.8.4.2407

[4] Perdana, A.A.M., Muhammad, F.B., Mappalotteng, A.M. (2025). Enhancing Batik classification leveraging CNN models and transfer learning. JOIV: International Journal on Informatics Visualization, 9(3): 1033-1040. https://doi.org/10.62527/JOIV.9.3.2535

[5] Budi, S., Affanti, T.B., Mataram, S. (2024). Ornamental patterns of contemporary Indonesian Batik: Clothing for strengthening the articulation of appearance characteristics. Wacana Seni Journal of Arts Discourse, 23: 16-28. https://doi.org/10.21315/ws2024.23.2

[6] Febriani, R., Knippenberg, L., Aarts, N. (2023). The making of a national icon: Narratives of Batik in Indonesia. Cogent Arts & Humanities, 10(1): 1-16. https://doi.org/10.1080/23311983.2023.2254042

[7] Wesnina, W., Prabawati, M., Noerharyono, M. (2025). Integrating traditional and contemporary in digital techniques: The analysis of Indonesian Batik motifs Evolution. Cogent Arts & Humanities, 12(1): 2474845. https://doi.org/10.1080/23311983.2025.2474845

[8] Kusumastuti, A., Atika, Achmadi, T.A., Phusavat, K., Hidayanto, A.N. (2023). Assessment of producer's perspective on the production of environmentally friendly fashion products: A case study in Indonesian natural dyes Batik craftsmen. Environmental Science and Pollution Research, 30: 124767-124779. https://doi.org/10.1007/s11356-022-23330-z

[9] Dzulqarnain, M.F., Fadlil, A., Riadi, I. (2025). Performance comparison of learned features from autoencoder and shape-based Hu moments for Batik classification. Jurnal Teknik Informatika, 6(4): 1729-1744. https://doi.org/10.52436/1.JUTIF.2025.6.4.4827

[10] Riadi, I., Fadlil, A., Purwadi Putra, I.J.D.E. (2023). Batik pattern classification using naïve bayes method based on texture feature extraction. Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika, 9(1). https://doi.org/10.23917/khif.v9i1.21207

[11] Winarno, E., Solichan, A., Ramdani, A.P., Hadikurniawati, W., Septiarini, A., Hamdani, H. (2025). Enhanced Semarang Batik classification using deep learning: A comparative study of CNN architectures. Bulletin of Electrical Engineering and Informatics, 14(5): 3544-3557. https://doi.org/10.11591/EEI.V14I5.9347

[12] Dzulqarnain, M.F., Fadlil, A., Riadi, I. (2024). Improving the accuracy of Batik classification using deep convolutional auto encoder. Compiler, 13(2): 123-130. https://doi.org/10.28989/compiler.v13i2.2649

[13] Fan, Y.Q., Hong, C.Q., Zeng, G.H., Liu, L.J. (2024). A deep convolutional encoder–decoder–restorer architecture for image deblurring. Neural Processing Letters, 56: 27. https://doi.org/10.1007/s11063-024-11455-w

[14] Farooq, Y., Savas, S. (2024). Noise removal from the image using convolutional neural networks-based denoising auto encoder. Journal of Emerging Computer Technologies, 3(1): 21-28. https://doi.org/10.57020/ject.1390428

[15] Küçükyilmaz, E.S., Akın, E. (2025). Investigation of geometric invariance properties of Hu moments for image processing applications. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 16(3): 571-580. https://doi.org/10.24012/DUMF.1670505

[16] Oumarou, H., Rismayanti, N. (2024). Automated classification of empon plants: A comparative study using Hu moments and K-NN algorithm. Indonesian Journal of Data and Science, 4(3): 206-214. https://doi.org/10.56705/ijodas.v4i3.115

[17] Satoto, B.D., Agustiono, W., Hamid, S.B., et al. (2026). Classification of Batik patterns using Inception-ResNetV2 with data augmentation. Journal of Advances in Information Technology, 17(1): 42-54. https://doi.org/10.12720/JAIT.17.1.42-54

[18] Gancheva, I., Peneva, E. (2023). Methodology based on the Hu moment invariants for object comparison on radar satellite imagery. Journal of Physics: Conference Series, 2668: 012012. https://doi.org/10.1088/1742-6596/2668/1/012012

[19] Chuan, Z.L., Chong, D.T.W., Lee, C.W.Y., et al. (2023). A comparative of two-dimensional statistical moment invariants features in formulating an automated probabilistic machine learning identification algorithm for forensic application. Malaysian Journal of Fundamental and Applied Sciences, 19(4): 525-538. https://doi.org/10.11113/mjfas.v19n4.2917

[20] Li, J.Z., Wang, J., Lin, Z.X. (2024). SGCAST: Symmetric graph convolutional auto-encoder for scalable and accurate study of spatial transcriptomics. Briefings in Bioinformatics, 25(1): 1-10. https://doi.org/10.1093/bib/bbad490

[21] Zhao, R., Yang, Z.W., Meng, X.C., Shao, F. (2024). A novel fully convolutional auto-encoder based on dual clustering and latent feature adversarial consistency for hyperspectral anomaly detection. Remote Sensing, 16(4): 717. https://doi.org/10.3390/rs16040717

[22] Septiani, B., Hasanuddin, T., Astuti, W. (2025). Classification of Lontara script using K-NN algorithm, decision tree, and random forest based on Hu moments and canny segmentation. Indonesian Journal of Data and Science, 6(2): 163-174. https://doi.org/10.56705/IJODAS.V6I2.281
[23] Rodríguez, J.A.M. (2023). Micro-scale surface recognition via microscope system based on Hu moments pattern and micro laser line projection. Metals, 13(5): 889. https://doi.org/10.3390/met13050889
[24] Minarno, A.E., Fadhlan, M., Munarko, Y., Chandranegara, D.R. (2024). Classification of dermoscopic images using CNN-SVM. International Journal on Informatics Visualization, 8(2): 606-612. https://doi.org/10.62527/JOIV.8.2.2153
[25] Fauzi, H., Aurellia, S., Alia, F. (2026). A multi-representation hybrid CNN feature extraction framework for cervical pre-cancer image classification. International Journal of Online and Biomedical Engineering, 22(5): 185-204. https://doi.org/10.3991/ijoe.v22i05.59729
[26] Anggriandi, D., Utami, E., Ariatmanto, D. (2023). Comparative analysis of CNN and CNN-SVM methods for classification types of human skin disease. Sinkron: Jurnal Dan Penelitian Teknik Informatika, 7(4): 2168-2178. https://doi.org/10.33395/sinkron.v8i4.12831
[27] Suryawanshi, S., Patil, S.B. (2024). Efficient brain tumor classification with a hybrid CNN-SVM approach in MRI. Journal of Advances in Information Technology, 15(3): 340-354. https://doi.org/10.12720/jait.15.3.340-354
[28] Yasar, A. (2024). Analysis of selected deep features with CNN-SVM-based for bread wheat seed classification. European Food Research and Technology, 250: 1551-1561. https://doi.org/10.1007/s00217-024-04488-x
[29] Jabbooree, A.I., Khanli, L.M., Salehpour, P., Pourbahrami, S. (2023). Geometrical facial expression recognition approach based on fusion CNN-SVM. International Journal of Intelligent Engineering & Systems, 17(1): 457-468. https://doi.org/10.22266/ijies2024.0229.40
[30] Fadlil, A., Umar, R., Sunardi, Nugroho, A.S. (2022). Comparison of machine learning approach for waste bottle classification. Emerging Science Journal, 6(5): 1075-1085. https://doi.org/10.28991/ESJ-2022-06-05-011
[31] Sunardi, Riadi, I., Sugandi, A. (2019). Forensic analysis of docker swarm cluster using GRR rapid response framework. International Journal of Advanced Computer Science and Applications, 10(2): 459-466. https://doi.org/10.14569/IJACSA.2019.0100260
[32] Riadi, I., Umar, R., Busthomi, I., Muhammad, A.W. (2022). Block-hash of blockchain framework against man-in-the-middle attacks. Register: Jurnal Ilmiah Teknologi Sistem Informasi, 8(1): 1-9. https://doi.org/10.26594/register.v8i1.2190
[33] Riadi, I., Sunardi, Aprilliansyah, D. (2023). Analysis of Anubis Trojan attack on Android Banking application using mobile security labware. International Journal of Safety and Security Engineering, 13(1): 31-38. https://doi.org/10.18280/ijsse.130104
[34] Riadi, I., Umar, R., Firdonsyah, A. (2018). Forensic tools performance analysis on android-based blackberry messenger using NIST measurements. International Journal of Electrical and Computer Engineering, 8(5): 3991-4003. https://doi.org/10.11591/ijece.v8i5.pp3991-4003
[35] Laakom, F., Raitoharju, J., Iosifidis, A., Gabbouj, M. (2024). Reducing redundancy in the bottleneck representation of autoencoders. Pattern Recognition Letters, 178: 202-208. https://doi.org/10.1016/j.patrec.2024.01.013
[36] Han, M., Dang, Y., Han, J.D. (2024). Denoising and baseline correction methods for Raman spectroscopy based on convolutional autoencoder: A unified solution. Sensors, 24(10): 3161. https://doi.org/10.3390/S24103161
[37] Alya, R.F., Wibowo, M., Paradise, P. (2023). Classification of Batik motif using transfer learning on convolutional neural network (CNN). Jurnal Teknik Informatika, 4(1): 161-170. https://doi.org/10.52436/1.jutif.2023.4.1.564
[38] Spina, D.E., Campos, L.F.D.O., de Arruda, W.F., et al. (2024). Comparison of autoencoder architectures for fault detection in industrial processes. Digital Chemical Engineering, 12: 100162. https://doi.org/10.1016/J.DCHE.2024.100162
[39] Huq, A., Reza, M.T., Hossain, S., Dipto, S.M. (2024). AnoMalNet: Outlier detection based malaria cell image classification method leveraging deep autoencoder. International Journal of Reconfigurable and Embedded Systems, 13(1): 171-178. https://doi.org/10.11591/ijres.v13.i1.pp171-178