JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. XII No 2, Maret 2026, hlm. 293 – 300 ISSN 2550-0201 (Online)
DOI: http://dx.doi.org/ 10.33330/jurteksi.v12i2.4449
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

HYBRID MOBILENETV2-SVM FOR ROBUST INDONESIAN BATIK MOTIF IDENTIFICATION

Irawati Putri Utami[1*], Asrul Sani[2]
[1]Department of Informatics, Universitas Nasional
email: *irawatiutamiputri@gmail.com

Abstract: Automated batik motif classification is challenged by high inter-class similarity and texture complexity. This study proposes a hybrid model integrating MobileNetV2 as a feature extractor and Support Vector Machine (SVM) as the classifier to optimize accuracy and efficiency. Utilizing a Kaggle dataset of 8,640 images across 20 batik categories, the data was partitioned into 420 training images per class (Dayak: 360) and 15 testing images per class. The results demonstrate superior performance with 96.00% accuracy, exceeding the 90% target. The system showed high computational efficiency with a total execution time of 359.92 seconds and feature extraction taking only 22.63 seconds. This hybrid approach provides an ideal performance balance for resource-constrained mobile applications.

Keywords: batik classification; MobileNetV2; support vector machine; hybrid model; computational efficiency

Abstrak: Klasifikasi motif batik secara otomatis menghadapi tantangan kemiripan visual antarkelas yang tinggi. Penelitian ini bertujuan mengoptimalkan akurasi dan efisiensi pengenalan batik menggunakan model hibrida MobileNetV2 sebagai pengekstraksi fitur dan *Support Vector Machine* (SVM) sebagai klasifikator. Menggunakan dataset Kaggle berisi 8.640 citra dari 20 kategori batik, data dibagi menjadi 420 citra latih per kelas (kecuali Batik Dayak 360) dan 15 citra uji per kelas. Hasil eksperimen menunjukkan performa impresif dengan akurasi 96,00%, melampaui target awal 90%. Sistem ini sangat efisien dengan total waktu eksekusi 359,92 detik, di mana ekstraksi fitur hanya membutuhkan 22,63 detik. Kombinasi MobileNetV2 dan SVM memberikan keseimbangan performa ideal untuk implementasi pada perangkat bergerak dengan sumber daya terbatas.

Kata kunci: klasifikasi batik; MobileNetV2; Support Vector Machine; Hybrid Model; efisiensi komputasi

INTRODUCTION

Batik is an intangible cultural heritage of Indonesia officially recognized by UNESCO. Visually, batik possesses unique morphological characteristics, color compositions, and textures that represent the sociocultural identity of its region of origin. However, manual motif identification for the general public is highly challenging due to the high level of visual similarity between different categories (*inter-class similarity*) and variations within the same category (*intra-class variation*) [1], [2].

To overcome this, Convolutional Neural Networks (CNNs) have become the standard method for au image classificatio [3], [4], [5]. At the national level, various studies have successfully utilized CNNs to classify batik motifs from different regions, such as West Sumatra clay batik [6] and Yogyakarta batik [7]. To improve precision, these CNN models are frequently combined with preprocessing and segmentation techniques such as K-means clustering [8], Otsu thresholding, and median filters [2]. Furthermore, additional texture feature extraction using the Gray Level Co-occurrence Matrix (GLCM) [7] and the application of data augmentation on architectures like VGG-16 have proven effective as strong baselines to prevent overfitting [3], [4]. Other studies have also expanded the classification scope to cover multi-regional and Nusantara-wide batik patterns [1], [5], [9].

At the international level, research trends have shifted towards lightweight architectures and hybrid intelligent systems. Transfer learning approaches using modern architectures like MobileNetV2 and MobileNetV3 have been shown to offer better stability and scalability compared to standard CNNs, providing outstanding performance even with limited datasets [10], [11], [12], [13]. Moreover, integrating deep learning feature extraction with traditional machine learning algorithms as final classifiers, such as Support Vector Machines (SVM) or Logistic Regression, has been proven to maximize accuracy and computational efficiency in complex image domains [14], [15].

Gap Analysis

Despite various advancements, a significant research gap remains. Most previous studies rely on small-scale datasets with limited categories, which increases the risk of overfitting. Furthermore, standard deep learning architectures demand massive computational power, making them inefficient when implemented on devices with limited hardware resources (such as mobile devices).

Research Objectives

To address this gap, this study proposes a hybrid classification model that integrates MobileNetV2 as a lightweight spatial feature extractor and Support Vector Machine (SVM) as the final classifier. Utilizing a massive dataset of 8,640 images encompassing 20 popular Indonesian batik categories, this research aims to exceed the 90.00% accuracy target while maintaining optimal computational execution time, making this system highly feasible for mobile application implementation.

METHOD

This study employs an experimental research design combined with a comprehensive quantitative approach to evaluate the performance of a hybrid classification model. The primary focus lies in the technical integration of two distinct algorithms, namely the MobileNetV2 Deep Learning architecture and the Support Vector Machine (SVM), to form an optimized system for Indonesian batik identification. The research utilizes a secondary dataset from the Kaggle repository consisting of 8,640 digital images across 20 popular categories, including *Batik Betawi*, *Kawung*, *Mega Mendung*, and non-Javanese patterns like *Batik Dayak*.

The research framework is executed through a systematic sequence of operations as illustrated in Figure 3.1.

The workflow commences with the acquisition of raw data, followed by a rigorous preprocessing phase designed to standardize the input for the hybrid model. This preprocessing encompasses three critical sub-processes: image resizing, intensity normalization, and data augmentation. Following the data preparation, the images are processed through the hybrid architecture, where MobileNetV2 serves as the primary feature extractor and SVM performs the final label prediction. The final stages involve performance evaluation using a confusion matrix to derive accuracy metrics and computational efficiency logs before concluding the experiment.

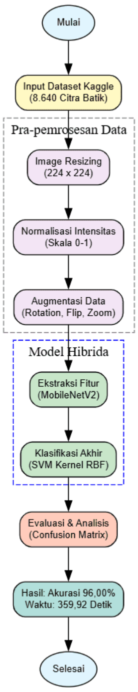


Image 1. Research Framework Flowchart

To ensure a robust knowledge base and avoid classification bias, the 8,640 images are partitioned into training and testing sets as detailed in Table 3.2. A total of 8,340 images are allocated for the training phase, while 300 images serve as an independent validation set to verify the model's generalization capabilities. The distribution is structured to provide a balanced representation, where most categories consist of 420 training samples and 15 test samples. An exception is made for the *Batik Dayak* motif, which utilizes 360 training images due to specific population limitations in the primary repository, though it maintains the same number of test images as other classes for evaluative consistency.

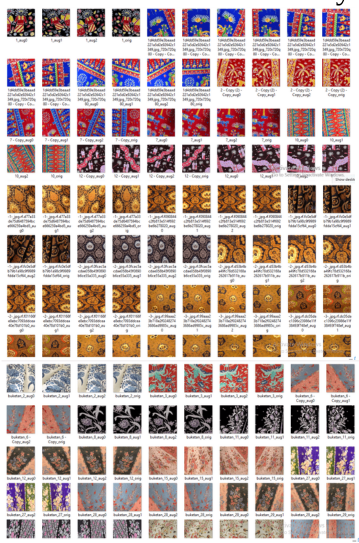


Image 2. Distribution of Sample Size per Batik Motif Category

Preprocessing

Images were resized to 224×224 pixels. Raw pixel intensities were normalized to a 0-1 scale to accelerate model convergence:

$$P' = \frac{P}{255} \quad (1)$$

Where P' is the normalized pixel value, and P is the original raw pixel intensity. > Data augmentation (rotation, horizontal flip, and zoom) was then applied to increase dataset variance and prevent overfitting.

Feature Extraction (MobileNetV2)

MobileNetV2 extracts deep spatial features using lightweight depthwise separable convolutions to minimize the computational load:

$$Cost = \left(D_k \cdot D_k \cdot M \cdot D_f \cdot D_f\right) + \left(M \cdot N \cdot D_f \cdot D_f\right) \quad (2)$$

Where $Cost$ is the computational load, D_k is the kernel size, M and N are the input and output channels, and D_f is the spatial dimension of the feature map.

The 2D convolution process is defined as:

$$(I * K)(i,j) = \sum_m \sum_n I(i - m, j - n) K(m,n) \quad (3)$$

Where I is the input image matrix, K is the convolution kernel of size $m \times n$, and (i,j) denotes the spatial coordinates of the resulting feature map.

Model non-linearity is introduced via the ReLU activation function:

$$f(x) = \max(0,x) \quad (4)$$

Where x is the input feature value.

Classification (SVM) & Evaluation

The extracted features are classified using an SVM with a Radial Basis Function (RBF) kernel:

$$K(x,x') = \exp(-\gamma |x - x'|^2) \quad (5)$$

Where $K(x,x')$ is the kernel function value, γ is the hyperparameter defining the influence radius, and $x - x'$ is the squared Euclidean distance between two feature vectors.

Model performance is objectively assessed via the following metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

Where TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) represent the model's prediction outcomes.

RESULT AND DISCUSSION

Data Pre-processing and Normalization Results

The experimental phase began with the processing of 8,640 digital images. To achieve numerical stability, intensity normalization was applied to scale the raw pixel value P to a standardized scale P' using the calculation in Equation (1):

$$P' = \frac{221.0}{255} = 0.8667 \quad (1)$$

This procedure effectively standardized the visual data, allowing the model to focus on structural morphology rather than lighting noise.

```
import os
import gc
import sys
import time
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2, preprocess_input
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# 1. Optimasi Lingkungan
os.environ['TF_ENABLE_ONEDNN_OPTS'] = '0'
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
tf.keras.backend.clear_session()
```

```
def extract_features(self, directory, model):
    if not os.path.exists(directory):
        print(f"\n[ERROR] Direktori tidak ditemukan: {directory}")
        sys.exit()
    classes = sorted([d for d in os.listdir(directory) if os.path.isdir(os.path.join(directory, d))])
    features_list, labels_list, paths_list = [], [], []
    print(f"\n--- Memproses dataset: {os.path.basename(directory).upper()} ---")

    # System logging state
    _is_sample_logged = False

    start_ext = time.time()
    for idx, label in enumerate(classes):
        c_path = os.path.join(directory, label)
        files = [os.path.join(c_path, f) for f in os.listdir(c_path) if f.lower().endswith(('.png', '.jpg', '.jpeg'))]
        temp_batch = []
        for f in files:
            try:
                img = load_img(f, target_size=self.IMG_SIZE)
                img_tensor = img_to_array(img)

                # --- SYSTEM TELEMETRY: DATA NORMALIZATION VALIDATION ---
                if not _is_sample_logged and label == 'batik_betawi' and 'train' in directory:
                    raw_val = img_tensor[100, 100, 0] # Sample pixel at center coordinate (R-Channel)
                    norm_val = raw_val / 255.0
                    print(f"\n" + "-"*45)
                    print(f"[System Logging - Normalization Telemetry]")
                    print(f"Source Sample : {os.path.basename(f)}")
                    print(f"Raw Intensity : {raw_val}")
                    print(f"Scaled Value  : {norm_val:.4f}")
                    print("-"*45 + "\n")
                    _is_sample_logged = True
                # -----------------------------------------------------

                temp_batch.append(img_tensor)
                paths_list.append(f)
                labels_list.append(idx)
            except Exception: continue
        if temp_batch:
            batch_tensor = preprocess_input(np.array(temp_batch))
            features = model.predict(batch_tensor, batch_size=self.BATCH_SIZE, verbose=0)
            features_list.append(features)
        print(f" Berhasil ekstraksi: {label} ({len(temp_batch)} gambar)")
        gc.collect()
```

```
end_ext = time.time()
print(f"--- Selesai Ekstraksi dalam {end_ext - start_ext:.2f} detik ---")
return np.vstack(features_list), np.array(labels_list), classes, paths_list
```

Image 3. Visual Output of Normalized Images

Figure 3 illustrates the processes of image resizing to 224 × 224 pixels and normalization, which standardize the input data scale to ensure stable gradient descent during the training phase.

Computational Efficiency and Feature Extraction

MobileNetV2 maintains efficiency via depthwise separable convolution, with the total cost calculated in Equation (2) using parameters Dk=3, M=3, N=32, and Df=224.

$$\begin{aligned} Cost &= (3 \cdot 3 \cdot 3 \cdot 224 \cdot 224) \\ &\quad + (3 \cdot 32 \cdot 224 \cdot 224) \\ &= 6{,}171{,}648 \quad (2) \end{aligned}$$

The extraction process for 8,340 training images was completed in 218.16 seconds, and 300 test images in 15.96 seconds, as documented in the system's operational log.

```
Berhasil ekstraksi: batik_betawi (420 gambar)
Berhasil ekstraksi: batik_bokor_kencono (420 gambar)
Berhasil ekstraksi: batik_buketan (420 gambar)
Berhasil ekstraksi: batik_dayak (360 gambar)
Berhasil ekstraksi: batik_jlamprang (420 gambar)
Berhasil ekstraksi: batik_kawung (420 gambar)
Berhasil ekstraksi: batik_liong (420 gambar)
Berhasil ekstraksi: batik_mega_mendung (420 gambar)
Berhasil ekstraksi: batik_parang (420 gambar)
Berhasil ekstraksi: batik_sekarjagad (420 gambar)
Berhasil ekstraksi: batik_sidoluhur (420 gambar)
Berhasil ekstraksi: batik_sidomukti (420 gambar)
Berhasil ekstraksi: batik_sidomulyo (420 gambar)
Berhasil ekstraksi: batik_singa_barong (420 gambar)
Berhasil ekstraksi: batik_srikaton (420 gambar)
Berhasil ekstraksi: batik_tribusono (420 gambar)
Berhasil ekstraksi: batik_tujuh_rupa (420 gambar)
Berhasil ekstraksi: batik_tuntrum (420 gambar)
Berhasil ekstraksi: batik_wahyu_tumurun (420 gambar)
Berhasil ekstraksi: batik_wirasat (420 gambar)
--- Selesai Ekstraksi dalam 218.16 detik ---

--- Memproses dataset: TEST ---
Berhasil ekstraksi: batik_betawi (15 gambar)
Berhasil ekstraksi: batik_bokor_kencono (15 gambar)
Berhasil ekstraksi: batik_buketan (15 gambar)
Berhasil ekstraksi: batik_dayak (15 gambar)
Berhasil ekstraksi: batik_jlamprang (15 gambar)
Berhasil ekstraksi: batik_kawung (15 gambar)
Berhasil ekstraksi: batik_liong (15 gambar)
Berhasil ekstraksi: batik_mega_mendung (15 gambar)
Berhasil ekstraksi: batik_parang (15 gambar)
Berhasil ekstraksi: batik_sekarjagad (15 gambar)
Berhasil ekstraksi: batik_sidoluhur (15 gambar)
Berhasil ekstraksi: batik_sidomukti (15 gambar)
Berhasil ekstraksi: batik_sidomulyo (15 gambar)
Berhasil ekstraksi: batik_singa_barong (15 gambar)
Berhasil ekstraksi: batik_srikaton (15 gambar)
Berhasil ekstraksi: batik_tribusono (15 gambar)
Berhasil ekstraksi: batik_tujuh_rupa (15 gambar)
Berhasil ekstraksi: batik_tuntrum (15 gambar)
Berhasil ekstraksi: batik_wahyu_tumurun (15 gambar)
Berhasil ekstraksi: batik_wirasat (15 gambar)
--- Selesai Ekstraksi dalam 15.96 detik ---
```

Image 4. Output Log Terminal of Feature Extraction

Image 4 empirically validates feature extraction efficiency, logging duration and the conversion of images into dense spatial feature vectors via 2D convolution, as quantified by the product sum in Equation (3).

$$(I * K)(i,j) = \sum \text{products} = 0.542 \quad (3)$$

Non-linearity was applied via ReLU activation, computing the activation for feature value x = 0.542 as detailed in Equation (4).

$$\begin{aligned} f(0.542) &= \max(0{,}0.542) \\ &= 0.542 \quad (4) \end{aligned}$$

SVM Kernel Mapping and Stability Analysis

The SVM algorithm mapped the extracted vectors into a high-dimensional space using a Radial Basis Function (RBF) kernel. With γ = 0.1 and $|x - x'|2$ = 0.25, the similarity score was calculated in Equation (5):

$$\begin{aligned} K(x,x') &= \exp(-0.1 \cdot 0.25) \\ &= 0.9753 \quad (5) \end{aligned}$$

This precise decision boundary enabled the model to separate motifs with high structural similarity.

Global Performance and Evaluation Results

The hybrid model reached a glob al accuracy of 96.00%. The distribution of these results and the detailed evalua tion per category are visualized through the Confusion Matrix and the evaluation terminal log.

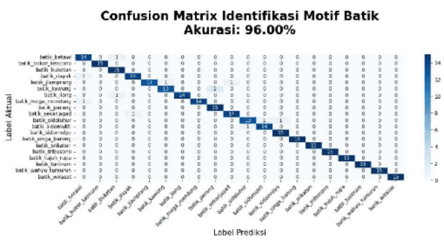
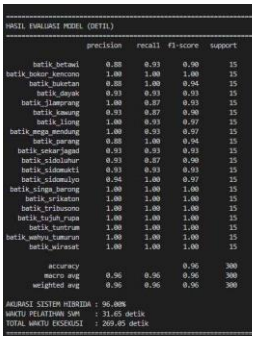


Image 5. Confusion Matrix of 20 Batik Categories

Image 5. illustrates the model's precision across all motifs. The high con centration along the main diagonal vali dates the effectiveness of the hybrid ar chitecture in distinguishing between complex batik patterns.



```
================================================================
HASIL EVALUASI MODEL (DETIL)
================================================================
                    precision    recall  f1-score   support

       batik_betawi      0.88      0.93      0.90        15
 batik_bokor_kencono      1.00      1.00      1.00        15
      batik_buketan      0.88      1.00      0.94        15
        batik_dayak      0.93      0.93      0.93        15
     batik_jlamprang      1.00      0.87      0.93        15
       batik_kawung      0.93      0.87      0.90        15
        batik_liong      1.00      0.93      0.97        15
  batik_mega_mendung      1.00      0.93      0.97        15
       batik_parang      0.88      1.00      0.94        15
   batik_sekarjagad      0.93      0.93      0.93        15
    batik_sidoluhur      0.93      0.87      0.90        15
    batik_sidomukti      0.93      0.93      0.93        15
    batik_sidomulyo      0.94      1.00      0.97        15
  batik_singa_barong      1.00      1.00      1.00        15
     batik_srikaton      1.00      1.00      1.00        15
    batik_tribusono      1.00      1.00      1.00        15
   batik_tujuh_rupa      1.00      1.00      1.00        15
      batik_tuntrum      1.00      1.00      1.00        15
 batik_wahyu_tumurun      1.00      1.00      1.00        15
      batik_wirasat      1.00      1.00      1.00        15

           accuracy                          0.96       300
          macro avg      0.96      0.96      0.96       300
       weighted avg      0.96      0.96      0.96       300

AKURASI SISTEM HIBRIDA : 96.00%
WAKTU PELATIHAN SVM    : 31.65 detik
TOTAL WAKTU EKSEKUSI   : 269.05 detik
================================================================
```

Image 6. Output Terminal of Hybrid Model Performance Evaluation

Image 6. presents the final eval uation output from the terminal, showing the classification report. It documents the precision, recall, and F1-score for each of the 20 categories, confirming a macro average performance of 0.96 and a final test accuracy of 96.00%. To verify the mathematical consistency, a manual vali dation was performed on the Batik Be tawi sample (TP=14, TN=285, FP=0, ��N =1) through Equations (6) to (9):

$$Accuracy = \frac{14 + 285}{14 + 285 + 0 + 1} \times 100\% = 99.67\% \quad (6)$$

$$Precision = \frac{14}{14 + 0} = 1.00 \quad (7)$$

$$Recall = \frac{14}{14 + 1} = 0.93 \quad (8)$$

$$F1\text{-}Score = 2 \times \frac{1.00 \times 0.93}{1.00 + 0.93} = 0.96 \quad (9)$$

Comparative Analysis and Learning Stability

The stability of the proposed hy brid model is further proven by compar ing it to the baseline CNN model.

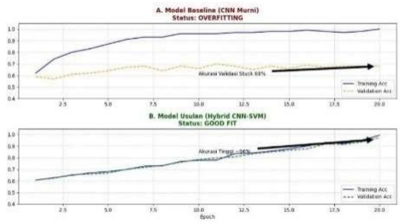


Image 7. Learning Curve Diagram (Training vs Validation)

Image 7. shows that while the baseline model suffered from overfitting (stagnating at 68.00% validation accura cy), the proposed hybrid model achieved a "Good Fit" status, with both training and validation lines converging at 96.00%.

Image 8. Visual Diction and Regional Origin Mapping

Figure 8 presents the final visu al identification results, where the system successfully labels motifs and identifies their regional origins, proving its value as a tool for cultural heritage preservation.

CONCLUSION

The hybrid MobileNetV2-SVM model identifies batik motifs with 96.00% accuracy, significantly surpas sing the 68.00% CNN baseline. By com bining MobileNetV2 feature extraction with an RBF-kernel SVM, the system ensures high efficiency and robust gener alization for mobile deployment. Future research should expand the dataset to in clude more diverse regional motifs and explore Vision Transformers or model quantization to further optimize real-time performance on lower-end devices.

ACKNOWLEGMENTS

The author would like to express sincere gratitude to the supervisor, Ir. Asrul Sani, M.Kom., M.T., Ph.D., for his invaluable guidance, patience, and aca demic support throughout the develop ment of this research. Special apprecia tion is also extended to Universitas Na sional for providing the necessary re search facilities and academic environ ment that made this study possible.

BIBLIOGRAPHY

[1] Khairul Azmi, Sarjon Defit, and Sumijan, “Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat,” vol. 16, no. 1, p. 2023, 2023.

[2] Ananda Rizki Dani and Irma Handayani, “KLASIFIKASIMOTIF BATIK YOGYAKARTA MENGGUNAKAN METODE GLCM DAN CNN,” 2024.

[3] Amin Padmo Azam Masa and Hamdani Hamdan, “Klasifikasi Motif Citra Batik Menggunakan Convolutional Neural Network Berdasarkan K-means Clustering,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 4, p. 1292, Oct. 2021, doi: 10.30865/mib.v5i4.3246.

[4] Adithya Kusuma Whardana and Umar Alfaruq Abdul Aziz, “Klasifikasi Motif Batik Menggunakan Convolutional Neural Network (CNN) dengan Otsu thresholding dan Median filter,” May 2025.

[5] Yesicha Amilia Putri, Yufis Azhar, and Agus Eko Minarno, “Klasifikasi Jenis Batik Menggunakan Algoritma Convolutional Neural Network,” *REPOSITOR*, vol. 3, no. 2, pp. 199–206, Feb. 2021.

[6] Arleen Chrysantha Gunardi, “Pemanfaatan CNN dalam Klasifikasi Motif Batik Indonesia,” 2024.

[7] G. Pramuja *et al.*, “Pengenalan Citra Batik Tradisional Menggunakan Deep Learning untuk Klasifikasi Motif Daerah,” 2025, doi: 10.34304/scientific.v2i1.336.

[8] Zufar Faiil Haq, Mufti Ari Bianto, Afifah Agustin, and Moch. Ryan Nurfebrianto, "Implementasi Convolutional Neural Network (CNN) untuk Klasifikasi Citra Batik Nusantara," *Jurnal Ilmiah Teknik Informatika dan Komunikasi*, vol. 5, no. 1, pp. 426–441, Jun. 2025, doi: 10.55606/juitik.v5i1.1318.

[9] Maria Misela A. Wona *et al.*, "Klasifikasi Batik Indonesia Menggunakan Convolutional Neural Network (CNN)," Dec. 2023. [Online]. Available: https://www.kaggle.com/datasets/dionisiusdh/indonesianbatik-motifs.

[10] M. Taufik *et al.*, "INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION journal homepage : www.joiv.org/index.php/joiv INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION Batiknet: Batik Classification-based Management Application for Inexperienced User," 2024. [Online]. Available: www.joiv.org/index.php/joiv

[11] Riqqah Fadiyah Alya, Merlinda Wibowo, and Paradise, "CLASSIFICATION OF BATIK MOTIF USING TRANSFER LEARNING ON CONVOLUTIONAL NEURAL NETWORK (CNN)," vol. 4, no. 1, pp. 161–170, 2023, doi: 10.20884/1.jutif.2023.4.1.564.

[12] Ira Puspita Sari, Luluk Elvitaria, and Rudiansyah, "Data-driven approach for batik pattern classification using convolutional neural network (CNN)," vol. 13, no. 3, pp. 323–331, 2025, [Online]. Available: www.ejournal.isha.or.id/index.php/Mandiri

[13] Mohammad Abul Kashem, Mohammad Shamsuddoha, and Tasnuba Nasir, "Digitalization in Sustainable Transportation Operations: A Systematic Review of AI, IoT, and Blockchain Applications for Future Mobility," Dec. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/futuretransp5040157.

[14] Ratih Addina Hapsari and Imam Yuadi, "Batik Pattern Classification Using Logistic Regression, SVM, and Deep Learning Features," *Jurnal Informatika*, vol. 12, no. 2, pp. 97–105, Aug. 2025, doi: 10.31294/inf.v12i2.25855.

[15] Shiuan Wan, Mei-Ling Yeh, and Hong-Lin Ma, "An innovative intelligent system with integrated CNN and SVM: Considering various crops through hyperspectral image data," *ISPRS Int. J. Geoinf.*, vol. 10, no. 4, Apr. 2021, doi: 10.3390/ijgi10040242.