Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 3543~3554
e-ISSN: 2548-6861

Comparative Analysis of MobileNetV2, Xception, and EfficientNet for Batik Pattern Classification

Maxentia Kathleen [1], Christy Atika Sari [2*]
Department of Informatics Engineering, Universitas Dian Nuswantoro, Indonenesia
111202315443@mhs.dinus.ac.id [1], christy.atika.sari@dsn.dinus.ac.id

Article Info

Article history:

Received 2026-07-10
Revised 2026-07-28
Accepted 2026-08-10

Keyword:

Batik,
Convolution Neural Network,
MobileNetV2,
Xception,
EfficientNet.

ABSTRACT

Indonesia boasts a rich cultural heritage in the form of batik, which has gained international recognition. However, the wide variety of batik motifs makes visual identification difficult for both locals and tourists. The limitations of manual observation and a lack of understanding regarding the significance behind each design pose a significant barrier to cultural preservation in the digital age. This study aims to conduct a comparative analysis of Deep Learning models to identify the most effective architecture for automatically classifying batik motifs. The method employed involved comparing three Convolutional Neural Network architectures: MobileNetV2, Xception, and EfficientNet. This study was conducted using a dataset containing 3,700 batik images that had been processed through a careful data distribution process. The primary objective of this evaluation is to find the optimal balance between high classification accuracy and efficient use of computational resources, enabling implementation on platforms with limited specifications. The results of this study indicate that these models can recognize complex batik patterns with outstanding validation accuracy rates ranging from 98% to 100%. These findings provide a strong technical foundation for selecting the most appropriate model architecture for developing intelligent systems aimed at preserving traditional batik. This study also shows that MobileNetv2 is the most optimal model architecture because it achieves a perfect balance between 100% accuracy and the fastest total inference time of 36.74 seconds, with an average inference time per sample of 0.0525 seconds. It is hoped that this research will make the batik identification process faster, more accurate, and accessible to the general public, thereby supporting the sustainability of Indonesia's cultural heritage.

I. INTRODUCTION

Indonesia is an archipelago comprising diverse ethnic groups, cultures, and languages. One of the cultural traditions deeply rooted in Indonesia is batik fabric. Batik is a vital aspect of Indonesia's cultural heritage that is recognized worldwide and designated as a UNESCO (United Nations Educational, Scientific and Cultural Organization) intangible cultural heritage of humanity[1]. Batik is an art form known for its traditional production method, which involves drawing repeating patterns onto fabric using a tool called a canting. Batik itself features a wide variety of unique motifs originating from the regions where it is produced [2]. This serves as a distinctive characteristic and identity of the existing batik motifs. To ensure batik does not disappear and continues to thrive, everyone must unite to protect it. Therefore, it is crucial to consistently preserve and maintain batik [3].

These designs represent a fusion of art, history, and identity that showcases the rich heritage of each region [4]. The vast diversity of batik motifs makes it difficult to distinguish designs across different types of batik. This poses a unique challenge for both locals and tourists in distinguishing and recognizing these motifs [3]. The difficulties currently faced by both local residents and tourists in identifying batik motifs significantly hinder the process of digitally preserving batik.

Until now, batik recognition has relied solely on manual visual observation, which is often limited by time and location and prone to errors. This is due to the fact that traditional batik is handmade, with each region's distinctive motifs created through complex, manual processes. However, the human eye has limitations in distinguishing and recognizing batik motifs in today's era. There is a critical need to develop an intelligent system that [5]. Understanding batik helps us recognize that these motifs serve not only as visual decorations but also as a vital medium for conveying the traditions and customs of the community [6].

One method for improving the recognition of traditional batik patterns is to develop an intelligent system capable of identifying these patterns and providing relevant information. One promising approach is Deep Learning, a branch of Machine Learning that uses layers to extract important information from raw data. In image pattern recognition, algorithms such as Convolutional Neural Network (CNN) are frequently used. Convolutional Neural Network (CNN) are highly effective at processing image data, including distinguishing objects from one another. One reason why Convolutional Neural Network (CNN) are popular is their ability to handle more complex datasets without requiring a separate feature extraction process, as feature extraction is performed internally during model development[7]. Convolutional Neural Network (CNN) are biologically inspired neural network architectures capable of learning hierarchical representations of raw image data, ranging from low-level features (edges, corners, textures) to high-level semantic abstractions (pattern structures or thematic identities). Its layered architecture, consisting of convolutional, pooling, and fully connected layers, enables Convolutional Neural Network (CNN) to extract, combine, and refine visual features with remarkable precision [8]. Although this research has various limitations and requires significant computational power, many of these approaches are impractical for real-world applications. Furthermore, some studies have not been sufficient to address the challenge of limited dataset availability, which is common in the field of cultural heritage. Variations in the quality and consistency of batik images further complicate classification, especially when dealing with traditional patterns that may exhibit subtle variations. Several studies have combined Convolutional Neural Network (CNN) architectures to improve batik classification. These architectures were selected based on their structural characteristics and respective advantages [9]. Previous research on batik classification has largely focused on accuracy while inadequately addressing computational efficiency for resource-constrained edge devices. This study addresses that gap by conducting a comprehensive comparative analysis of lightweight models. This study proposes using the MobileNetV2, Xception, and EfficientNet model architectures. These three architectural models were selected because they are the latest lightweight CNN architectures designed to achieve an optimal balance between parameter efficieny and high classification performance, making them suitable for intelligent systems [10]. The MobileNetV2 architecture has been used to classify batik patterns, achieving validation accuracy of up to 100%. Its implementation in an EfficientNet-based model also yielded validation accuracy of up to 100%. Meanwhile, the Xception-based model achieved a validation accuracy of 98%. These findings demonstrate the ability of Deep Learning models to recognize complex batik patterns. However, many studies have focused more on classification accuracy, while attention to computational efficiency, convergence stability, and optimization behavior remains limited. These technical aspects are crucial for real-world applications, particularly on platforms with limited resources [11].

Based on the issues discussed and previous research, an evaluation is needed to identify the most effective deep learning model for classifying batik patterns. The primary focus of this study is to achieve a balance between classification accuracy and resource efficiency. This evaluation was conducted using a dataset consisting of 3,700 batik images with an accurate data distribution. This study comprehensively compares the performance of the MobileNetV2, Xception, and EfficientNet architectures. The results of this evaluation are expected to determine which architecture is most suitable for classifying traditional batik motifs.

II. METHOD

A. Literature Review

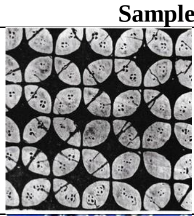
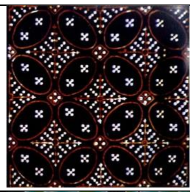
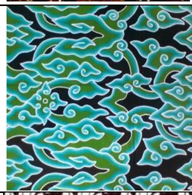
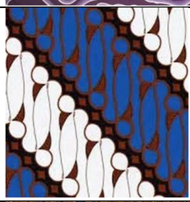
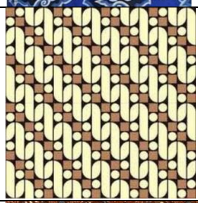
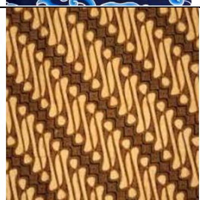
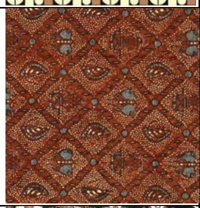
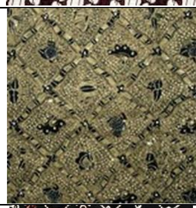
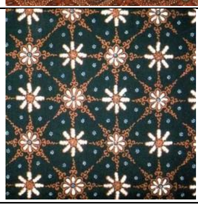
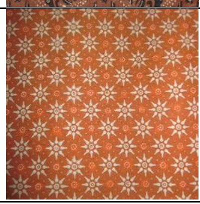
Batik is a craft with high cultural value that has been passed down by the ancestors of the Indonesian people since ancient times. The art of batik-making has flourished and evolved, producing a wide variety of batik motifs. The technique of batik making is an indigenous skill that has been practiced by the Indonesian people for centuries. The development of batik styles has also been influenced by local environments, leading to a growing interest in batik among the Indonesian public. This is evident in the increasing use of batik in everyday attire, both formal and informal. Beyond clothing, various jewelry and decorative items also feature numerous batik motifs. Therefore, the preservation of batik must be sustained to ensure its continued existence [12]. The development of image classification systems has seen rapid progress, particularly with the integration of deep learning approaches. Convolutional Neural Network (CNN) are widely used in image recognition tasks due to their ability to automatically learn hierarchical visual representations directly from raw image inputs. Convolutional Neural Network (CNN) architectures themselves are widely applied across various fields, such as agricultural image classification, facial expression recognition, and cultural heritage analysis, demonstrating strong resilience in handling complex visual patterns and textures. This study confirms the effectiveness of Convolutional Neural Network (CNN) in capturing subtle inter-class differences, emphasizing improved accuracy while not analyzing convergence stability a critical factor when dealing with visually similar classes and limited datasets.

Several studies have examined Convolutional Neural Network (CNN)-based classification specifically for batik motifs. This highlights the primary general limitation of deeper Convolutional Neural Network (CNN) models for batik classification, particularly for applications in resource-constrained systems where efficiency and convergence speed are as important as accuracy [11], [13], [14]. Therefore, this research aims to preserve culture and learning technology.

B. Data Collection

The data used in this study consists of a collection of Coastal Batik (Cirebon) and Royal Batik (Central Java / Yogyakarta) obtained from GitHub (https://github.com/arifnurrhmnn/Dataset-Motif-Batik.git). This study uses publicly available datasets to support the reproducibility of the research result. This dataset was selected due to the cultural and historical significance of the Coastal Batik (Cirebon) and Royal Batik (Central Java / Yogyakarta) motifs. These motifs possess distinctive visual characteristics and are rich in meaningful symbolism, making them suitable for preserving Indonesia's heritage through computational analysis [16]. There are five types of batik motif classes to be used in the data collection process for each image dataset: Batik Kawung, Batik Megamendung, Batik Parang, Batik Sidomukti, and Batik Truntum. These images are in JPG format and feature popular and easily identifiable batik motifs. Each type of object is placed in a class folder labeled according to its type. The dataset used in this study consists of batik motifs categorized into five groups: Batik Kawung, Batik Megamendung, Batik Parang, Batik Sidomukti, and Batik Truntum. This dataset has been divided into three subsets: training, validation, and testing.

TABLE I
SAMPLE IMAGES BY CLASS FOR EACH BATIK MOTIF

Batik Motif Name	Sample of dataset			
Batik Kawung				
Batik Megamendung				
Batik Parang				
Batik Sidomukti				
Batik Truntum				

The five batik motifs were divided into training, validation, and testing sets. This data split resulted in a 70:15:15 ratio, yielding 700 samples for training, 15 images for validation, and 15 images reserved as unseen data. This ensures that the sample size distribution across the five motifs maintains a relatively fair proportion. This study identifies five categories of batik motifs to be used in the data collection process for each image dataset are Kawung Batik, Megamendung Batik, Parang Batik, Sidomukti Batik, and Truntum Batik. Each class of batik motifs is shown in Table I. The figure illustrates the visual characteristics and complexity of the geometric patterns that form the basis for spatial feature extraction by the Convolutional Neural Network (CNN) architecture. This reduces the likelihood of texture bias toward a specific dominant class and ensures an evaluation matrix that accurately reflects the model across all motif categories, as illustrated in Table II. The diversity of textures, dominant ornamental shapes, and intricate details in the five batik motifs provide a sufficient level of intra-class and inter-class feature variability to test the image recognition generalization capability of the model. Next, to assess and mitigate the risk of overfitting regarding weight convergence, the entire dataset was processed using a hold-out validation framework.

TABLE II
FEATURE CLASSES CIVIDED INTO TRAINING, VALIDATION, AND TESTING SETS

Motif Class	Training	Validation	Testing
Batik Kawung	560	140	140
Batik Megamendung	560	140	140
Batik Parang	560	140	140
Batik Sidomukti	560	140	140

C. Data Preprocessing

Data preprocessing is a crucial step in preparing the dataset for training Convolutional Neural Network (CNN). This study resized each image in the dataset to 224 x 224 pixels, in accordance with the input size required by the MobileNetV2, Xception, and EfficientNet model architectures [17] as in Figure 1.

```
train_data = train_gen.flow_from_directory(
    DATASET_PATH + '/train',
    target_size=(224,224),
    batch_size=32,
    class_mode='categorical'
)

val_data = val_gen.flow_from_directory(
    DATASET_PATH + '/val',
    target_size=(224,224),
    batch_size=32,
    class_mode='categorical'
)

test_data = test_gen.flow_from_directory(
    DATASET_PATH + '/val',
    target_size=(224,224),
    batch_size=1,
    class_mode='categorical',
    shuffle=False,
    classes=list(train_data.class_indices.keys())
)
```

Figure 1. Code for reading the dataset for the training, validation, and test data

The selection of the 224 x 224-pixel size is based on the specifications of the MobileNetV2, Xception, and EfficientNet models, which have been trained using datasets with images of the same size. Resizing images to a uniform size offers several key benefits, including ensuring input compatibility with the models and reducing computational load, as smaller images can be processed more efficiently. Therefore, preserving color information in the images is important to support classification accuracy [10].

D. Data Splitting

After preprocessing, the dataset was divided into three separate subsets: train, validation, and test, with a ratio of 70% for training, 15% for validation, and 15% for testing. The 70:15:15 split ratio was chosen based on empirical studies of similar pattern recognition tasks, which provide sufficient data for training while ensuring reliable validation and testing as illustrated in Figure 2. Data splitting is performed by first shuffling the images within each class. This is important to avoid bias that might arise if the data were sorted according to specific patterns. After the shuffling process, the image files are copied to the appropriate folders, ensuring that the folder structure is preserved and each image is treated separately during dataset splitting. This step ensures that each model is trained on data representative of each class, tested on separate data, and validated using data not used during training [10]. Data collection through the splitting process serves a research purpose aimed at improving the model's ability to identify batik patterns based on the images in the dataset. Data collection allocated for validation serves to assess the model during the training process and helps reduce the risk associated with overfitting. Conversely, data collection designated for testing is used to evaluate the overall model's ability to categorize previously unseen images. Through the implementation of appropriate dataset partitioning, model performance evaluation can be conducted with improved precision.

```
train_counts = count_class_distribution(train_data, "train")
val_counts   = count_class_distribution(val_data, "val")
test_counts  = count_class_distribution(test_data, "test")
```

Figure 2. Class labels for the training, validation, and test datasets

E. Data Augmentation

To improve the model's generalization ability and reduce the risk of overfitting caused by the limited size of the training dataset, data augmentation was applied exclusively to the training set using the ImageDataGenerator framework. The goal of data augmentation is to increase data variability by applying controlled transformations that preserve the semantic characteristics of the original batik motifs while exposing the model to a wider range of visual patterns. This approach is particularly crucial for classifying detailed batik, where subtle variations in texture, orientation, and lighting significantly influence model learning. The augmentation techniques used in this study include random rotation, horizontal shift, vertical shift, scaling, horizontal flipping, and brightness adjustment with nearest-neighbor interpolation applied to fill in empty pixel areas generated by geometric transformations [18]. This study was chosen to simulate real-world variations in batik photography, such as differences in camera angle, scale, and lighting conditions. However, the validation set was intentionally excluded from augmentation to maintain an unbiased evaluation of model performance [11]. Only normalized rescaling was applied to validation images to ensure consistency with the input distribution expected by the MobileNetV2, Xception, and Efficient models.

F. Model Training

The next step is model training, where data is loaded and preprocessed before being used for training, such as splitting the dataset into training and validation sets, resizing, normalizing, and augmenting the data. Hyperparameters such as the number of epochs, learning rate, batch size, and optimizer are selected. Afterward, the training process proceeds through each epoch, including the forward pass, backward pass, and weight updates [19]. Figure 3 illustrates that the batik pattern images were standardized to 224 x 224 pixels to align with the specifications of the neural network architecture to be used. The training process was then limited to 100 epochs, utilizing a batch size of 32 images per iteration to manage the computational load on memory resources. Network weight optimization is performed using the Adam optimizer algorithm, starting with an initial learning rate of 0.0001 to evaluate prediction errors within the classification framework and the cross-entropy loss function used. Furthermore, to reduce overfitting, the dropout regularization technique with a rate of 0.2 is applied to the fully connected layers. In the network's output layer, the Softmax activation function is used to generate probabilistic values corresponding to each class, thereby facilitating the determination of the final classification result. In this study, model training was conducted using Convolutional Neural Network (CNN) models as pre-trained models. The Convolutional Neural Network (CNN) models used as pre-trained models include MobileNetV2, Xception, and EfficientNet. To ensure a fair comparison, all three architectures were trained and evaluated under completely identical experimental conditions, including the same 224 x 2224 input size, a similar dataset split (70:15:15), and the same hyperparameter configurations. MobileNetV2: The first model used in this study is MobileNetV2, which is based on a lightweight and efficient architecture. This framework is carefully engineered to minimize the number of parameters while accelerating the computational process. During the training phase, the MobileNetV2 model enables the system to identify batik motifs, including linear shapes, curves, and textures. The benefits of MobileNetV2 include accelerated training, reduced model size, compatibility with low-spec devices, and reduced computational load [20]. Xception : The second model used in this study is the Xception model, as it is an extension of the Inception architecture. It employs a comprehensive deep separate convolutional approach, enabling the model to generate more detailed features and process data optimally. The process in this model is divided into two stages: depth and point convolutions, which allow Xception to extract more detailed and complex visual information from images. Its advantages include faster training times, improved computational efficiency, and a reduced number of required parameters [21]. EfficientNet: The third model used in this study is the EfficientNet model, which is a family of Convolutional Neural Network (CNN) designed to achieve high performance with fewer computational resources compared to previous architectures. The EfficientNet model has excellent feature extraction capabilities. It has fewer parameters and higher accuracy compared to other traditional Convolutional Neural Network (CNN). The image size used is 224 x 224 pixels. To build a basic EfficientNet network, a multi-objective neural architecture search approach was used, which was then scaled simultaneously in depth, width, and resolution to achieve an optimal balance [8].

	Hyperparameter	Value
0	Image Size	(224, 224)
1	Batch Size	32
2	Epoch	100
3	Optimizer	Adam
4	Learning Rate	0.0001
5	Loss Function	categorical_crossentropy
6	Dropout	0.2
7	Activation Function	Softmax

Figure 3. Proposed Hyperparameter

G. Research Stages

Within the scope of this research, we collected and analyzed batik pattern motifs and identified appropriate data sources and types. The primary objective is to develop a model capable of recognizing and classifying batik motifs through image processing. The proposed model utilizes a Convolutional Neural Network (CNN) that involves several stages to extract and analyze the distinctive features of batik patterns [3]. This research began with data collection, data preprocessing, dataset splitting, data augmentation, model training, and evaluation, as shown in Figure 4.

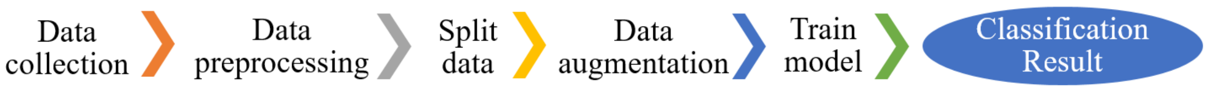


Figure 4. Research Flowchart

H. Evaluation Model

The evaluation process involves testing the model using the data to be evaluated. Next, accuracy is calculated by determining how many correct predictions there are compared to the total number of predictions generated. The accuracy result provides an overview of the model's ability to recognize patterns learned from the training data. For a more in-depth analysis, a confusion matrix is used, which shows a comparison between the true labels and the labels predicted by the model. Through the confusion matrix, we can identify the classes where the model frequently makes errors. The classification report includes metrics such as accuracy, precision, recall, and F1-Score, which will be calculated [10], [22]. Accuracy is used to measure the percentage of correct predictions from the entire test data set asin (1). Precision measures how accurately the model predicts a specific class, specifically by comparing the proportion of correct positive predictions to the total number of positive predictions as in (2). Recall is used to measure a model's ability to identify all true positive examples in the data asin (3). F1-Score is serves as the harmonic mean of precision and recall, which is used as a measure of the model's performance balance as in (4), where True Positive (TP) is the number of positive data points that the model correctly predicted as positive. True Negative (TN) is the number of negative data points that the model correctly predicts as negative. False Positive (FP) is the number of negative data points that the model incorrectly predicts as positive. False Negative (FN) is the number of positive data points that the model incorrectly predicts as negative.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

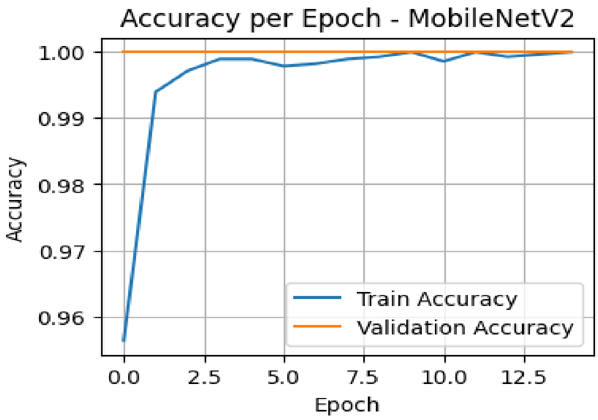
$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

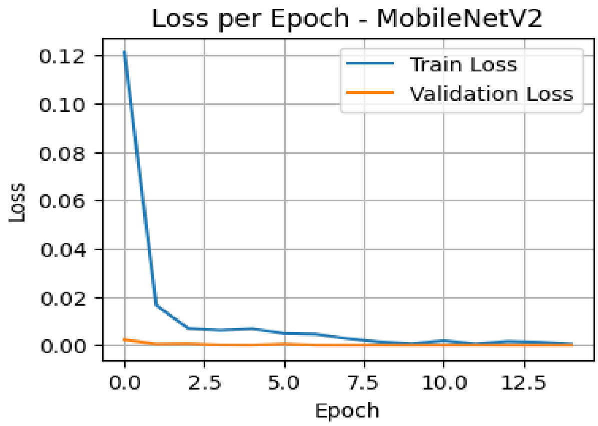
III. RESULTS AND DISCUSSION

A. Training model using MobileNetV2

In this study, the MobileNetV2 model was adapted for the extraction of visual representations of batik motifs through an analysis of the learning curve. In accordance with the experimental setup, the base MobileNetV2 model was created statically using pre-trained weights from ImageNet to ensure that the training process focused on modifying the weights in the terminal classification layer, which consists of Dense and Dropout layers. The accuracy and loss functions during the training period are illustrated in Figure 5. The Accuracy per Epoch graph above shows that the MobileNetV2 model is undergoing a very rapid convergence trajectory.



Graphic Accuracy using MobileNetV2



Graphic Loss using MobileNetV2

Figure 5. Accuracy and Loss per Epoch for MobileNetV2

Notably, for several epochs, the validation accuracy is slightly higher than the training accuracy. This can be attributed to the application of the Dropout regularization technique at rates of 50% and 20% in the hidden layers. Throughout training, the neural network systematically deactivates subsets of neurons to reduce over-reliance on specific features, which ultimately leads to a reduction in training accuracy metrics. Conversely, during validation, the entire architecture functions cohesively (ensemble effect), resulting in accurate predictions. Then, on the Accuracy curve, it starts with stability and reaches a maximum value for validation accuracy. This demonstrates a commendable capacity for generalization against variations in the classification patterns of previously unknown batik motifs. The Loss per Epoch graph shows that the MobileNetV2 model is experiencing a consistent decrease in both training loss and validation loss. This graph indicates a lack of substantial divergence, demonstrating that the MobileNetV2 model has been protected from overfitting. The 100% validation accuracy is not an indication of overfitting, as it is strictly supported by the stable convergence of the validation loss curve, the implementation of 50% and 20% dropout layers, and the hold-out validation method on unseen data.

During the training process configured for 100 epochs, training stopped automatically at the 15th epoch. The termination of iterations was executed by the EarlyStopping callback function after identifying the absence of a significant decrease in Validation Loss for 5 consecutive iterations (patience=5). This mechanism not only ensures that the architecture maintains an optimal weight configuration by restoring the best weights but also demonstrates the effectiveness of ReduceLronPlateau in dynamically reducing the learning rate when encountering local minima, thereby maintaining the convergence stability of the MobileNetV2 model.

The MobileNetV2 model was evaluated using test data to verify the classification of each different category of batik motifs. The results of this evaluation are presented in the confusion matrix shown in Figure 6. Analysis of the figure above shows that the MobileNetV2 model demonstrates excellent generalization capabilities, as evidenced by a negligible error rate. Out of a total of 170 test samples (the estimated sample size), the model accurately classified the majority of the motif categories. Specifically, the Batik Parang class achieved perfect accuracy with no classification errors. However, certain batik motifs exhibited similar textural characteristics: one Batik Kawung motif was classified as Batik Megamendung, and one Batik Sidomukti motif was classified as Batik Parang. This indicates a slight overlap in visual characteristics related to the defined geometric patterns and does not significantly impair the model's overall performance, as the accuracy metric remains at 100%. The MobilenetV2 model demonstrates highly consistent performance across various classes. In all categories, it achieves a perfect score (1.00) in all evaluation metrics, as shown in Table III. In the MobileNetV2 model, in addition to accuracy, computational efficiency is a key criterion in this study. The inference time results indicate that the MobileNetV2 model provides highly efficient computational performance.

The model requires only 36.74 seconds to process the entire test dataset, with an average inference time of 0.0525 seconds per sample as in Table IV. This significantly lower inference latency demonstrates that MobileNetV2 is well suited for real-time batik motif classification, particularly in resource-constrained or edge computing environments where fast prediction is essential.

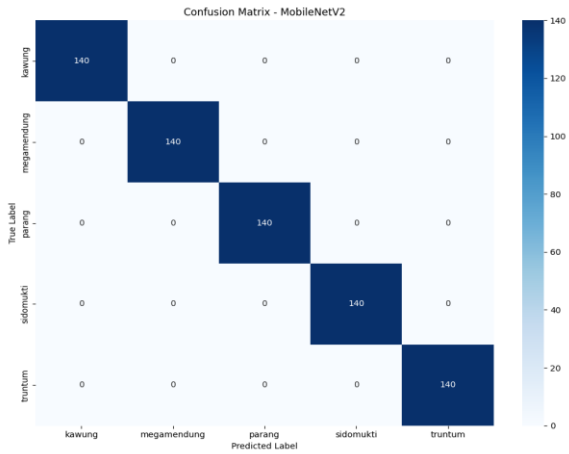


Figure 6. Confusion Matrix Model MobileNetV2

TABLE III
CLASSIFICATION REPORT MODEL MOBILENETV2

Motif Class	Precision	Recall	F1-Score	Support
Batik Kawung	1.00	1.00	1.00	140
Batik Megamendung	1.00	1.00	1.00	140
Batik Parang	1.00	1.00	1.00	140
Batik Sidomukti	1.00	1.00	1.00	140
Batik Truntum	1.00	1.00	1.00	140
Accuracy			1.00	700
Marco Avg	1.00	1.00	1.00	700
Weighted Avg	1.00	1.00	1.00	700

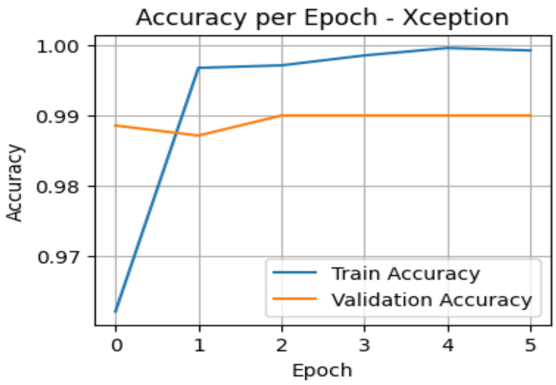
TABLE IV
INFERENCE TIME OF THE MOBILENETV2 MODEL

Category	Time
Total Inference Time	36.74 seconds
Average Inference Time per Sample	0.0525 seconds

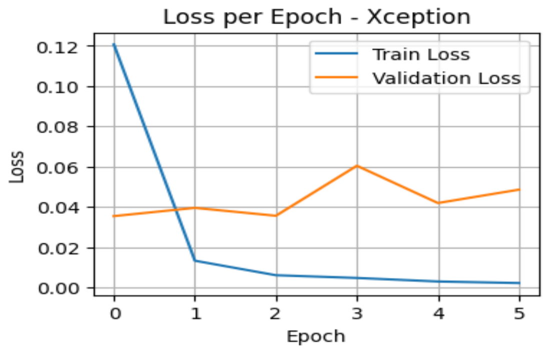
B. Training model using Xception

This study extends to the Xception model, which features a more complex architecture through the use of deep separated convolutions. The hyperparameter settings and regularization schemes applied to the Xception model remain consistent with previous experiments to ensure an objective comparison of performance. The learning dynamics of the Xception model throughout the training and validation phases are illustrated by the accuracy curves and loss functions shown in Figure 6.

The Accuracy per Epoch graph in Figure 7 illustrates that the Xception model demonstrates remarkable acceleration in visual feature learning during the early stages of training. Notably, the Validation Accuracy curve follows a nearly identical trajectory, closely aligning with the Train Accuracy curve from the fifth iteration through the end of training. The minimal gap observed between these curves serves as an empirical indicator showing that the representational capacity of the Xception model architecture is highly aligned with the level of complexity inherent in the batik motif dataset. The strategy of freezing ImageNet weights in the base layers and applying Dropout at rates of 50% and 20% in the final layers is effective in reducing the risk of overfitting.



Graphic Accuracy using Xception



Graphic Loss using Xception

Figure 7. Accuracy and Loss per Epoch for Xception

This allows the Xception model to achieve stable convergence accuracy. The Loss per Epoch graph illustrates that the Xception model provides deep insights into the dynamics associated with network weight optimization. Slight fluctuations in the Validation Loss observed during epochs five through eight highlight the challenges faced by the architecture in distinguishing batik features that exhibit intraclass similarity. However, the implementation of the ReduceLronPlateau callback proved to be crucial in this process; by dynamically adjusting the learning rate, the Adam optimization algorithm successfully mitigated the fluctuations and brought the model back to the gradient via EarlyStopping at the 17th epoch, validating that the model had found the most optimal global minimum without expending computational resources on epochs that did not yield significant metric improvements.

The confusion matrix in Figure 8 demonstrates that the proposed EfficientNetB4 model achieves excellent classification performance for the batik motif dataset. Most classes are classified correctly, with kawung, lereng, and sidomukti achieving 140 correct predictions, while megamendung and truntum achieve 139 and 133 correct predictions, respectively. Only a small number of misclassifications are observed, including one megamendung image misclassified as parang and seven truntum images misclassified as kawung. These results indicate that the model has high discriminative capability and can accurately distinguish between batik motifs, with only minor confusion between visually similar classes.

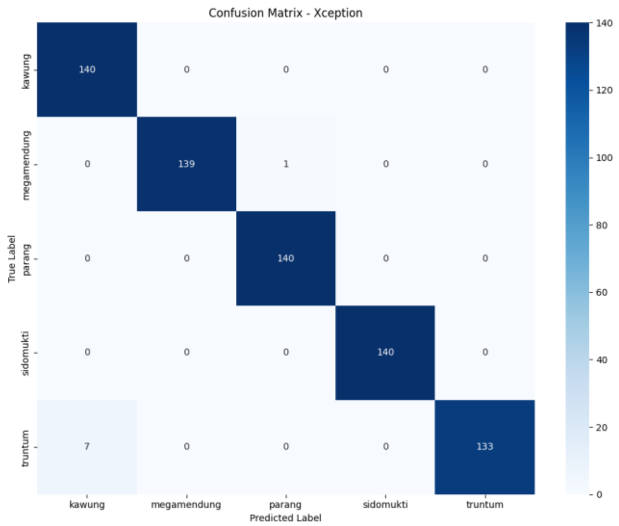


Figure 8. Confusion Matrix Model MobileNetV2

TABLE V
INFERENCE TIME OF THE XCEPTION MODEL

Category	Time
Total Inference Time	160.77 seconds
Average Inference Time per Sample	0.2297 seconds

The inference time analysis in Table V indicates that the proposed EfficientNet model is computationally efficient, requiring a total inference time of 160.77 seconds to process the entire test dataset. On average, the model classifies each sample in 0.2297 seconds, demonstrating its capability for fast and efficient image recognition. These results suggest that the proposed approach is suitable for practical batik motif classification applications where both high accuracy and low inference latency are important.

The classification results demonstrate that the proposed model achieves outstanding performance across all batik motif classes, with an overall accuracy of 99% as shown in Table VI. The model attains macro-average and weighted-average precision, recall, and F1-score of 0.99, indicating consistent performance across the balanced dataset. Among the five classes, Batik Sidomukti is classified perfectly with 100% precision, recall, and F1-score, while Batik Kawung and Batik Truntum show slightly lower recall values of 0.95, reflecting only a few misclassifications. Overall, these results confirm the robustness and reliability of the proposed model for batik motif classification.

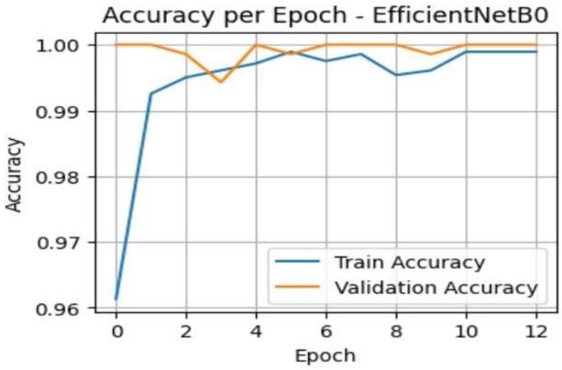
TABLE VI
CLASSIFICATION REPORT MODEL XCEPTION

Motif Class	Precision	Recall	F1-Score	Support
Batik Kawung	0.95	1.00	0.98	140
Batik Megamendung	1.00	0.99	1.00	140
Batik Parang	0.99	1.00	1.00	140
Batik Sidomukti	1.00	1.00	1.00	140
Batik Truntum	1.00	0.95	0.97	140
Accuracy			0.99	700
Marco Avg	0.99	0.99	0.99	700
Weighted Avg	0.99	0.99	0.99	700

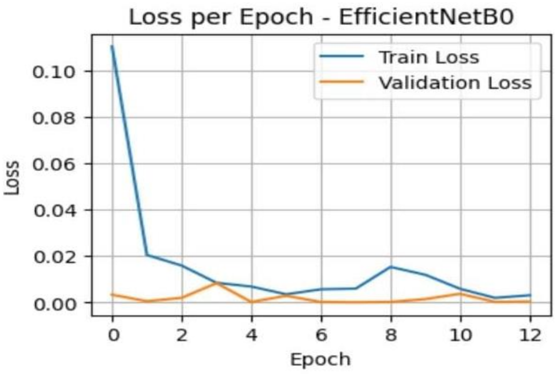
C. *Training model using EfficienNet*

This study extends to the EfficientNet model, which employs a multi-scale scaling methodology to optimize the network's dimensions in a quantifiable manner. In line with the experimental framework used for the MobileNetV2 and Xception models, the classification layer inherent to the model was replaced with a Global Average Pooling layer alongside two Dropout layers at 50% and 20% rates to ensure that the testing parameters remained consistent. The training of this architecture is graphically represented through the accuracy and loss curves illustrated in Figure 7.

The Accuracy per Epoch graph above illustrates that the EfficientNet model exhibits a highly progressive initial convergence phase. The training and validation curves show exponential growth in the early iterations, indicating the model's progress in mapping the visual feature hierarchy within batik pattern images. There is a notable absence of a widening gap between training and validation accuracy as the number of epochs increases. The convergence of these two curves serves as compelling empirical evidence that this model is resistant to overfitting and maintains a commendable balance in representing unseen data in Figure 10. The EfficientNet model achieves 100% validation accuracy scalability. The 100% validation accuracy is not an indication of overfitting, as it is strictly supported by the stable convergence of the validation loss curve, the implementation of 50% and 20% dropout layers, and the hold-out validation method on unseen data. The Loss per Epoch graph demonstrates that the model provides strong evidence of optimal convergence. The Validation Loss curve aligns with the Train Loss, showing no upward trend toward the end of the training. The weight optimization procedure performed by the Adam algorithm is highly effective, aided by the ReduceLronPlateau intervention, which successfully reduces loss fluctuations by adaptively lowering the learning rate when learning progress begins to plateau. The training process is automatically terminated by the EarlyStopping callback at the 14th epoch. This termination confirms that the EfficientNet architecture successfully extracts the most representative architectural parameters for batik pattern classification without compromising computational efficiency.



Graphic Accuracy using EfficienNet



Graphic Loss using EfficienNet

Figure 7. Accuracy and Loss per Epoch for EfficienNet

TABLE VII
INFERENCE TIME OF THE EFIICIENTNET MODEL

Category	Time
Total Inference Time	43.75 seconds
Average Inference Time per Sample	0.0625 seconds

The inference time evaluation in Table VII shows that the EfficientNet model achieves efficient computational performance while maintaining high classification accuracy. The model requires a total inference time of 43.75 seconds to process the entire test dataset, with an average inference time of 0.0625 seconds per sample. These results indicate that EfficientNet provides fast prediction speed, making it suitable for practical batik motif classification applications that require both high accuracy and low computational latency.

The classification report in Table VIII demonstrates that the Xception model achieves excellent performance, with an overall accuracy of 99% on the test dataset. The model attains macro-average and weighted-average precision, recall, and F1-score of 0.99, indicating balanced and consistent classification across all batik motif classes. Batik Sidomukti is classified perfectly with 100% precision, recall, and F1-score, while Batik Kawung and Batik Truntum show slightly lower recall values of 0.95, reflecting only a few misclassified samples. Overall, these results confirm that the Xception model is highly reliable and effective for batik motif classification.

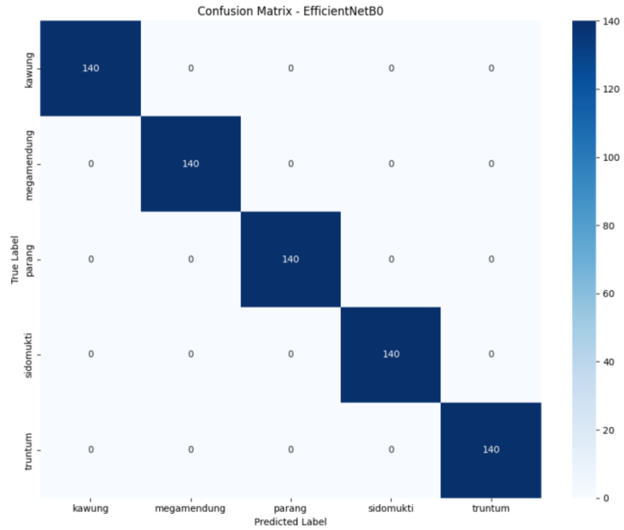


Figure 10. Confusion Matrix for the EfficientNet Model

TABLE VIII
CLASSIFICATION REPORT MODEL EFFICIENTNET

Motif Class	Precision	Recall	F1-Score	Support
Batik Kawung	1.00	1.00	1.00	140
Batik Megamendung	1.00	1.00	1.00	140
Batik Parang	1.00	1.00	1.00	140
Batik Sidomukti	1.00	1.00	1.00	140
Batik Truntum	1.00	1.00	1.00	140
Accuracy			1.00	700
Marco Avg	1.00	1.00	1.00	700
Weighted Avg	1.00	1.00	1.00	700

D. Comparison Result

A comparative analysis of evaluation metrics across the three evaluated deep learning architectures: MobileNetV2, Xception, and EfficientNet. This analysis was conducted using identical parameters with an input resolution of 224 x 224 pixels and applied a transfer learning methodology by freezing the base model to leverage pre-trained weights from ImageNet. Based on the above evaluation, the MobileNetV2 and EfficientNet models achieved perfect accuracy 100% with perfect Precision, Recall, and F1-Score, all recorded as 1.00. This indicates that MobileNetV2 and EfficientNet are highly proficient at extracting the intricate features inherent in batik patterns. Meanwhile, the Xception model achieved a validation accuracy of 98%, despite having a deeper architecture with deeply separable convolutions. Xception demonstrated a negligible margin of error in distinguishing specific batik pattern variations when compared to MobileNetV2 and EfficientNet. Although the results from the Xception model are the lowest among the three evaluated architectures, these results remain very high for a model engineered with a lightweight framework. Incorporating Global Average Pooling and Dropout layers into the code is highly effective for reducing the likelihood of overfitting and maintaining stability. In terms of computational efficiency, the MobileNetV2 architecture outperformed the others with the fastest total inference time of 26.74 seconds. In comparison, the EfficientNet architecture required a total inference time of 43.75 seconds, while the Xception architecture took the longest with a total inference time of 160.77 seconds.

Table IX describes a study was implemented Convolutional Neural Network (CNN) to classify batik patterns. The study conducted by D. A. Ramadhan [7] used a conventional Convolutional Neural Network (CNN) approach to classify Riau batik patterns via a website. Using a dataset of 1,440 images with various combinations of batch size and epoch, it achieved an accuracy of 89%. The study also combined data augmentation with a Confusion Matrix to evaluate the performance of the Riau batik pattern classification model. Meanwhile, the study conducted by E. Y. Puspaningrum [21] utilized the Xception model architecture, comparing activation functions on a South Sulawesi batik dataset. Using a dataset of 1,440 images, the study found that the ELU activation function achieved the highest accuracy of 98.57%. And the study conducted by M. Rafli [11] MobileNETv2 in conjunction with Bayesian Optimization and Particle Swarm Optimization (PSO) for hyperparameter tuning. This study focuses on the classification of Parang Solo and Parang Yogyakarta batik patterns using a dataset consisting of 160 distinct images.

TABLE IX
COMPARISON WITH PREVIOUS STUDIES

Title	Year	Method Differences	Dataset	Total Image	Accuracy (%)
D.A. Rahmadhan [23]	2024	Conventional Convolutional Neural Network (CNN) with optimized batch size and epoch settings, as well as a web-based implementation.	Batik Riau	1,400	89
E. Y. Puspaningrum [24]	2025	Us Xception architecture with improved activation functions ReLU, ELU, Leaky ReLU, and Swish to enhance classification performance.	South Sulawesi Batik (Lontara, Tongkonan, Pinisi, and Toraja motifs combined).	1,400	98.57
M. Rafli [25]	2026	Combining MobileNetV2 with Bayesian Optimization and PSO, and comparing the Adam and Adagrad optimizers to improve the stability and generalization of the tested model.	Batik Parang Solo and Yogyakarta.	160	91

Unlike previous studies, which primarily prioritized classification accuracy, this study also examines optimizer behavior by comparing the Adam and Adagrad optimizers to assess their convergence stability and generalization capabilities. Experimental findings show that the Adagrad optimizer, when integrated with Bayesian Optimization, achieves superior performance with an accuracy rate of 91%, implying that lightweight CNN architectures are capable of effectively classifying analog batik patterns visually while simultaneously maintaining computational efficiency. It can be concluded that previous research findings indicate that the application of Convolutional Neural Networks (CNNs) or transfer learning is highly effective for classifying batik patterns. However, some of these previous studies focused only on a single model architecture or specific parameters. Consequently, this study conducts a comparative analysis of several Convolutional Neural Network (CNN) models namely MobileNetV2, Xception, and EfficientNet to determine the most effective model for identifying batik motifs in Indonesia. This study also employs preprocessing, data augmentation, and evaluation methodologies using a Confusion Matrix to achieve more optimal and comprehensive classification performance.

IV. CONCLUSION

Based on research on the classification of batik motifs using deep learning models, it can be concluded that the application of transfer learning with a Convolutional Neural Network (CNN) has proven effective in recognizing the visual characteristics of various batik motifs. This study successfully demonstrated that each architecture exhibits different performance in terms of accuracy and computational efficiency. Based on the results of this study, MobileNetV2 is the best model for this application. Although both the MobileNetV2 and EfficientNet architectures achieved 100% accuracy and perfect evaluation metrics (Precision, Recall, and F1-Score of 1.00). The MobileNetV2 architecture offers better computational efficiency with the lowest inference time, making it the most optimal choice for deployment on platforms with limited resources. However, the Xception model demonstrates good performance with an accuracy rate of 98%. The evaluation results obtained through the Confusion Matrix reveal classification errors identified by the Xception model, which stem from similarities in the geometric patterns of certain motifs, such as Parang Batik and Megamendung Batik. However, the combination of dropout regularization and callback methods such as EarlyStopping and ReduceLronPlateau proved effective in maintaining convergence stability and preventing overfitting across all three models. Although this study still has a number of limitations, primarily because the dataset consists of images obtained under controlled conditions with a clean background. Consequently, the model's performance may decline when faced with real-world batik images captured under extreme lighting conditions or from inconsistent angles. Furthermore, this study covers only five classes of batik motifs, thereby limiting the model's ability to generalize across the full range of traditional patterns. Therefore, further research is strongly recommended to expand the dataset with a more diverse range of motif classes, apply more advanced data augmentation techniques, and conduct direct testing on edge computing devices to verify the model's robustness in dynamic real-world environments.

REFERENCES

[1] K. Kusnadi, "Exploring character education through batik Pekalongan local wisdom: An innovative approach to character learning," *Jurnal Civics: Media Kajian Kewarganegaraan*, vol. 20, no. 2, pp. 223–235, Oct. 2023, doi: 10.21831/jc.v20i2.57000.

[2] D. Gede, T. Meranggi, N. Yudistira, and Y. A. Sari, "Batik Classification Using Convolutional Neural Network with Data Improvements," *International Journal on Informatics Visualization*, Mar. 2022, doi: http://dx.doi.org/10.30630/joiv.6.1.716.

[3] A. Oktarino, Y. Desnita Tasri, and A. Efendi, "Identification of Batik Motif Based Deep Learning- Convolutional Neural Network

Approach," *Journal of Ocean, Mechanical and Aesrospace*, vol. 68, no. 3, Nov. 2024, doi: http://dx.doi.org/10.36842/jomase.v68i3.381.

[4] I. I'aza, T. Hosen, and S. A. Kamaruddin, "Batik Pattern Classification Using Machine Learning Approaches," *Applied Mathematics and Computational Intelligence*, 2024, doi: https://doi.org/10.58915/amci.v13i3.690.

[5] U. Muhdhor and Y. Yohannes, "Evaluation of MobileNet-Based Deep Features for Yogyakarta Traditional Batik Motif Classification," *Sinkron : Jurnal dan Penelitian Teknik Informatika*, vol. 10, no. 1, pp. 389–395, Jan. 2026, doi: 10.33395/sinkron.v10i1.15668.

[6] E. Utaminingsih and I. Sahputra, "Automated Recognition of Batik Aceh Patterns Using Machine Learning Techniques," *Brilliance: Research of Artificial Intelligence*, vol. 4, no. 2, pp. 619–624, Nov. 2024, doi: 10.47709/brilliance.v4i2.4831.

[7] D. A. Ramadhan and D. Ramadhani, "Classification of Riau Batik Motifs Using the Convolutional Neural Network (CNN) Algorithm," *International Journal of Electrical, Energy and Power System Engineering*, vol. 7, no. 3, pp. 201–211, Nov. 2024, doi: 10.31258/ijeepse.7.3.201-211.

[8] R. S. Putra, M. M. Al Haromainy, and A. Junaidi, "Classification of Jombang Batik Motifs Using Ensemble Convolutional Neural Network," *Bit-Tech (Binary Digital - Technology)*, vol. 8, no. 2, pp. 2102–2112, Dec. 2025, doi: 10.32877/bt.v8i2.3204.

[9] B. Sunarko *et al.*, "Identification of the Sub-motifs of Batik Kawung Using Deep Learning," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 25, no. 2, pp. 299–310, Mar. 2026, doi: 10.30812/matrik.v25i2.5818.

[10] E. Khoirunnisa *et al.*, "Enhanced Semarang Batik Classification using MobileNetV2 and Data Augmentation," *Sinkron : Jurnal dan Penelitian Teknik Informatika*, vol. 9, no. 1, pp. 43–54, Jan. 2025, doi: 10.33395/sinkron.v9i1.14308.

[11] M. Rafli, D. A. Prasetya, and K. M. Hindrayani, "Optimisation of Hyperparameter Tuning and Optimiser on MobileNetV2 for Batik Parang Classification," *Bit-Tech (Binary Digital - Technology)*, vol. 8, no. 3, pp. 3542–3552, Apr. 2026, doi: 10.32877/bt.v8i3.3576.

[12] A. Nurcahyo, N. Ishartono, A. Y. C. Pratiwi, and M. Waluyo, "Exploration Of Mathematical Concepts In Batik Truntum Surakarta," *Infinity Journal of Mathematics Education*, vol. 13, no. 2, pp. 457–475, Sep. 2024, doi: 10.22460/infinity.v13i2.p457-476.

[13] M. Idhom, D. A. Prasetya, P. A. Riyantoko, T. M. Fahrudin, and A. P. Sari, "Pneumonia Classification Utilizing VGG-16 Architecture and Convolutional Neural Network Algorithm for Imbalanced Datasets," *TIERS Information Technology Journal*, vol. 4, no. 1, pp. 73–82, Jun. 2023, doi: 10.38043/tiers.v4i1.4380.

[14] H. Sastypratiwi and H. Muhardi, "Batik Recognition and Classification Using Transfer Learning and MobileNet Approach," *International Journal on Informatics Visualization*, Dec. 2024, doi: http://dx.doi.org/10.62527/joiv.8.4.2407.

[15] L. A. Latumakulita, "Pattern Recognition of Puta Dino Fabric Using Web-Based Convolutional Neural Network Method," *Journal of Applied Data Sciences*, vol. 7, no. 2, pp. 1020–1035, May 2026, doi: 10.47738/jads.v7i2.1103.

[16] S. Suyahman and A. Hapsari, "VGG-Based Feature Extraction for Classifying Traditional Batik Motifs Using Machine Learning Models," *Preservation, Digital Technology and Culture*, vol. 54, no. 3, pp. 215–222, Oct. 2025, doi: 10.1515/pdtc-2025-0009.

[17] M. S. Huda, H. Endah Wahanani, and F. T. Anggraeny, "Optimizing the ResNet50 Model with Five Optimizers for Detecting Rice Leaf Diseases," *Bit-Tech (Binary Digital - Technology)*, vol. 8, no. 2, pp. 2285–2296, Dec. 2025, doi: 10.32877/bt.v8i2.3232.

[18] E. A. Nabila, C. A. Sari, E. H. Rachmawanto, and M. Doheir, "A Good Performance of Convolutional Neural Network Based on AlexNet in Domestic Indonesian Car Types Classification," *Advance Sustainable Science Engineering and Technology*, vol. 5, no. 3, p. 0230302, Oct. 2023, doi: 10.26877/asset.v5i3.16854.

[19] S. S. F. Ardyani and C. A. Sari, "A Web-Based for Demak Batik Classification Using VGG16 Convolutional Neural Network," *Advance Sustainable Science, Engineering and Technology*, vol. 6, no. 4, pp. 0240406-01-0240406–09, Aug. 2024, doi: 10.26877/asset.v6i4.771.

[20] A. Akbar, M. Perdana, M. Fajar, and A. Muis Mappalotteng, "Enhancing Batik Classification Leveraging CNN Models and Transfer Learning," *International Journal on Informatics Visualization*, May 2025, doi: http://dx.doi.org/10.62527/joiv.9.3.2535.

[21] E. Yulia Puspaningrum, B. Eden William Asrul, N. -Veteran, J. Timur, J. Adiyaksa Baru No, and J. Raya Rungkut Madya No, "Optimization of CNN Activation Functions using Xception for South Sulawesi Batik Classification," *Sistematis: Journal Sistem Informasi*, 2025, doi: https://doi.org/10.32520/stmsi.v14i5.5281.

[22] B. J. Filia *et al.*, "Improving Batik Pattern Classification using CNN with Advanced Augmentation and Oversampling on Imbalanced Dataset," in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 508–517. doi: 10.1016/j.procs.2023.10.552.