Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
19/1 (2026), 53-61. DOI: http://dx.doi.org/10.21609/jiki.v19i1.1597

Traditional Batik Pattern Recognition with MobileNetV2 and Sampling-Based Hyperparameter Optimization

Suyahman[1], Onesinus Saut Parulian[2], Deny Prasetyo[3], Muhammad Anwar Fauzi[4]

[1,3]Department of Computer Science, Universitas Sugeng Hartono, Sukoharjo, Indonesia
[2]Universitas Nusa Mandiri, Indonesia
[4]Department of Digital Business, Universitas Sugeng Hartono, Sukoharjo, Indonesia

E-mail: suyahman@sugenghartono.ac.id

Abstract

Batik holds significant cultural value in Indonesia, reflecting the nation's historical and artistic heritage through its intricate patterns. Preserving these designs is essential for maintaining cultural identity and supporting artistic and economic communities. With the advancement of technology, deep learning has emerged as an effective approach for recognizing and classifying batik patterns. Convolutional Neural Networks (CNNs), particularly MobileNetV2, are widely recognized for their efficiency and accuracy in image classification. However, the performance of deep learning models is highly influenced by hyperparameter selection, which remains a challenging task. This study investigates the effectiveness of MobileNetV2 in classifying traditional Indonesian batik motifs, including Kawung, Mega Mendung, Parang, and Truntum, by applying different hyperparameter optimization methods such as Tree-structured Parzen Estimator (TPE), Gaussian Process Sampler (GPS), Grid Search, and Random Search. The experimental results show that TPE achieved the best overall performance with a test accuracy of 91.94% and an F1 score of 92.09%. GPS and Grid Search obtained identical test accuracy of 90.83% with F1 scores of 90.89% and 90.87%, respectively, while Random Search produced the lowest performance with an accuracy of 88.61% and F1 score of 88.61%. These findings highlight the importance of structured hyperparameter optimization, particularly TPE, in enhancing the robustness of CNN-based batik classification. The results provide valuable insights for the development of automated batik pattern recognition systems that support cultural heritage preservation and related image classification applications.
Keywords: *Pattern Recognition, CNN, MobileNet, Hyperparameter Optimization, Batik*

1. Introduction

Batik is an essential part of Indonesia's cultural heritage, characterized by intricate patterns and distinctive motifs that reflect the nation's rich history and artistic traditions. As a symbol of identity and craftsmanship, preserving batik patterns is crucial not only for maintaining cultural heritage but also for supporting the economic and artistic communities involved in batik production [1]. The digitalization of batik pattern recognition and classification has gained increased relevance, facilitating automated preservation and documentation while enhancing accessibility for researchers, designers, and industries [2]. Despite its importance, the manual classification and analysis of batik patterns remain challenging due to the complexity and diversity of motifs, necessitating advanced computational approaches. Several studies have explored the application of deep learning approaches for batik classification and hyperparameter tuning in related domains.

Auliaddina & Arifin applied Convolutional Neural Networks (CNN) with Data Augmentation and Hyperparameter Tuning for batik classification, achieving 66.67% accuracy compared to 28.15% without these optimizations [3]. Their research highlighted that Data Augmentation had a more significant impact on accuracy improvement than Hyperparameter Tuning, increasing validation accuracy to 64% from the baseline 28.15%. Roland et al. approached batik pattern recognition by developing a CNN model automatically designed and optimized using genetic algorithms [4]. Their model achieved an accuracy of 86.54% while requiring only approximately 1% of the parameters used by the VGG-19 model, which achieved 75.96% accuracy. This demonstrates the potential of evolutionary algorithms in optimizing neural network architectures for batik classification. In related work, Murinto et al. compared CNN and Transfer Learning models for batik motif classification, specifically implementing a Particle

Swarm Optimization CNN (PSOCNN) with VGG16 transfer learning [5]. Their model achieved 83% accuracy, representing a 6% improvement over standard CNN approaches, highlighting the benefits of transfer learning and optimization techniques in this domain. Beyond batik classification, Vo et al. investigated hyperparameter tuning in transfer learning for driver drowsiness detection, comparing Bayesian optimization and Random search algorithms [6]. They examined various hyperparameters including dropout rate, activation function, number of dense nodes, optimizer, and learning rate across MobileNetV2, Xception, and VGG19 architectures. Their findings indicated that Bayesian optimization was more efficient than Random search for finding optimal hyperparameters.

Deep learning has emerged as a powerful tool for image classification tasks, including batik pattern recognition. Convolutional Neural Networks (CNNs) have been widely adopted for feature extraction and classification due to their superior ability to recognize intricate visual patterns [7]. Among various CNN architectures, MobileNetV2 has gained attention for its efficiency and effectiveness in image classification tasks. MobileNetV2 employs depthwise separable convolutions and inverted residual connections, significantly reducing computational costs while maintaining high accuracy. Its lightweight architecture makes it particularly suitable for mobile and embedded applications, providing an optimal balance between performance and efficiency in batik classification [8]. Despite the advancements in deep learning-based batik classification, achieving optimal performance heavily depends on the careful selection of hyperparameters.

Hyperparameter tuning is a crucial but challenging aspect, as improper configurations can lead to suboptimal model performance [9]. Parameters such as learning rate, batch size, number of layers, and dropout rates significantly influence the convergence and generalization capabilities of the model [10, 11, 12, 13, 14]. Traditional manual tuning approaches are inefficient and often require extensive trial-and-error experiments, highlighting the need for automated hyperparameter optimization techniques [15, 16, 17]. To address these challenges, various sampling-based algorithms have been proposed for efficient hyperparameter tuning. Techniques such as Bayesian optimization, grid search, random search, and evolutionary algorithms have been explored in different domains to enhance model performance while reducing computational costs. These methods systematically explore the hyperparameter space, identifying optimal configurations that improve accuracy and generalization [12,13]. The integration of these advanced optimization techniques offers significant potential for improving the accuracy and efficiency of batik pattern classification models.

This study aims to further explore the potential of MobileNetV2 for batik pattern classification while addressing the challenges of hyperparameter optimization using sampling-based search algorithms. Additionally, this study utilizes a dataset consisting of a collection of batik keraton motif images, which include four traditional Indonesian motifs: Kawung, Mega Mendung, Parang, and Truntum. These motifs hold significant cultural value and exhibit unique geometric and organic patterns that pose classification challenges. The dataset serves as a representative benchmark for evaluating the effectiveness of deep learning-based batik pattern classification models. By systematically evaluating different hyperparameter tuning techniques, this research seeks to improve classification accuracy and computational efficiency, contributing to the advancement of automated batik pattern recognition. The findings of this study are expected to provide insights into optimizing deep learning models for cultural heritage preservation and similar image classification tasks. Additionally, this study focuses on specific sampling methods, including TPESampler, GPSampler, GridSampler, and RandomSampler. These methods provide different trade-offs between exploration and exploitation in hyperparameter search, and their effectiveness in batik classification remains an area of investigation.

2. Methods

This study adopts a comprehensive approach to classifying Batik Keraton motif images by leveraging deep learning techniques and hyperparameter optimization. Figure 1 illustrates the research methodology applied.

The dataset consists of 1,799 images representing four distinct Batik motifs: Kawung, Mega Mendung, Parang, and Truntum. To enhance consistency and robustness, images undergo preprocessing steps such as resizing, normalization, and data augmentation. Hyperparameter tuning is conducted using both Bayesian (TPE, GP) and non-Bayesian (Grid, Random) sampling methods to determine optimal model configurations, focusing on learning rate, dropout rate, batch size, and dense unit count. MobileNetV2, pre-trained on ImageNet and fine-tuned for Batik classification, serves as the model's backbone. The training process incorporates feature extraction, global average pooling, dense layer processing, and softmax classification, optimized with the Adam optimizer alongside an early stopping mechanism. Model performance is rigorously evaluated on unseen data using metrics such as accuracy, precision, recall, F1-score, and a confusion matrix, ensuring a comprehensive assessment of its effectiveness in classifying Batik Keraton motifs.

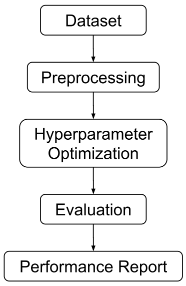

Figure 1. Flowchart of research methodology of this research.

2.1. Dataset

The dataset utilized in this study comprises images of Batik Keraton motifs, featuring four traditional Indonesian batik patterns: Kawung, Mega Mendung, Parang, and Truntum. It is structured to facilitate research in pattern recognition, image classification, and cultural preservation using advanced technologies such as computer vision and machine learning. Figure 2 is an example of Batik from this dataset.

Figure 2. Examples of Batik Kawung (a), Mega Mendung (b), Parang (c), and Truntum (d).

Stored in JPG format with the RGB color model, the dataset ensures compatibility with deep learning frameworks that process full-color images. A total of 1,799 images are included, distributed as follows: 506 images of Kawung, 472 of Mega Mendung, 426 of Parang, and 395 of Truntum. The dataset, sourced from Kaggle (https://www.kaggle.com/datasets/stefaron/dataset-batik-keraton), has been carefully curated to provide a diverse and well-organized collection of batik pattern samples.

2.2. Preprocessing

Before training the model, the dataset undergoes a preprocessing phase to ensure consistency and suitability for classification tasks. Each image in the dataset, originally in JPG format and RGB color space, is resized to a standard dimension of 128×128 pixels to ensure uniform input dimensions, balancing computational efficiency with adequate detail for feature extraction. The images are normalized by scaling pixel values to a range of [0,1], enhancing model convergence during training. Each image is retained as a three-channel RGB array, preserving the crucial color information necessary for distinguishing batik patterns.

To mitigate overfitting and enhance model generalization, data augmentation techniques are employed, including random rotation, horizontal flipping, and brightness adjustments. These augmentations simulate real-world variations, helping the model become more resilient to changes in lighting, orientation, and slight distortions [14]. The dataset is subsequently split into training (80%) and testing (20%) subsets, with stratified sampling ensuring a balanced distribution of each class. The labels are one-hot encoded to align with the categorical output format required for multi-class classification.

2.3. Hyperparameter Optimization

To determine the optimal hyperparameters for the classification model, various sampling algorithms are employed, categorized into Bayesian and non-Bayesian approaches. Each method has distinct characteristics in exploring and exploiting the hyperparameter space, influencing the efficiency and performance of the optimization process.

Bayesian optimization techniques utilize probabilistic models to guide the selection of hyperparameter values, thereby improving search efficiency. The Tree-structured Parzen Estimator (TPE) Sampler is an adaptive sampling algorithm that constructs two probability distributions: one representing promising hyperparameter values and another for less promising values. It selects new configurations by maximizing the expected improvement over the best-performing trial. This sampling algorithm effectively balances exploration, which involves testing new hyperparameter values, and exploitation, which refines previously successful values. TPE is particularly advantageous in high-dimensional and complex search spaces, making it well-suited for deep learning applications [15].

The Gaussian Process (GP) Sampler models the objective function using a Gaussian Process, estimating the relationship between hyperparameters and model performance [16]. It employs an acquisition function to guide the selection of new hyperparameter configurations by predicting which samples are likely to improve model accuracy. While effective for low-dimensional spaces, GP-based sampling can become computationally expensive as the number of hyperparameters increases.

Non-Bayesian sampling algorithms explore the hyperparameter space without relying on probabilistic models, making them simple to implement but often less efficient in identifying optimal configurations. Grid Search (GridSampler) systematically evaluates all possible hyperparameter combinations within a predefined set [17]. While this exhaustive approach ensures no configuration is overlooked, it becomes computationally impractical in large search spaces due to the exponential increase in required evaluations. Grid Search is most suitable when the number of hyperparameters is small or when the search space is well-defined.

In contrast, Random Search (RandomSampler) selects hyperparameter values randomly within specified ranges [18]. Although it does not guarantee the best solution, it often outperforms Grid Search in high-dimensional spaces by exploring a more diverse set of configurations within a limited number of trials. Its simplicity and effectiveness in identifying good hyperparameter values without exhaustive searching make it a widely used baseline in optimization tasks. To fine-tune key hyperparameters that significantly impact model performance, this study employs these sampling algorithms. The learning rate, chosen from a log-uniform range of 1e-5 to 1e-2, influences training stability and convergence speed. The dropout rate, sampled between 0.2 and 0.5, mitigates overfitting by randomly deactivating neurons, enhancing model generalization. Batch sizes of 16, 32, or 64 affect computational efficiency and gradient stability, while the number of dense units, selected from 64, 128, or 256, determines the complexity of fully connected layers after feature extraction. The optimization process runs for 20 trials, with each trial training the model and assessing validation accuracy. The configuration yielding the highest validation accuracy is chosen for final training.

While non-Bayesian methods provide a straightforward approach, Bayesian techniques, particularly TPE, are preferred for their efficiency in navigating the hyperparameter space. By focusing on promising regions, Bayesian optimization enhances classification performance, making it a more effective strategy for model tuning.

2.4. Training

The core of the classification model is MobileNetV2, an efficient yet robust convolutional neural network (CNN) architecture optimized for image classification tasks. Pre-trained on ImageNet, MobileNetV2 offers strong feature extraction capabilities, which are leveraged through transfer learning. Figure 3 provides an overview of MobileNetV2.

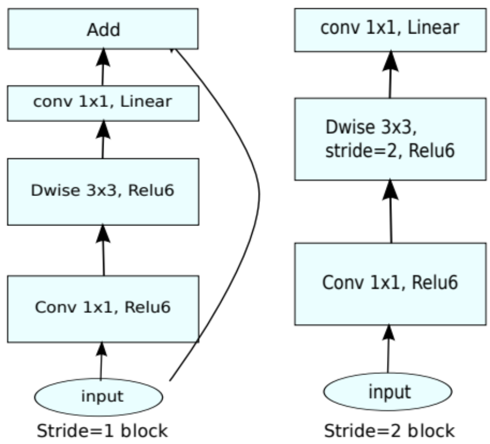

Figure 3. MobilenetV2 architecture [19].

In this method, the lower convolutional layers of the model remain unchanged to retain their general feature extraction abilities, while the upper layers are fine-tuned to recognize batik-specific patterns. This approach allows the model to benefit from pre-learned features while adapting to the unique characteristics of batik motifs.

The training process begins with feature extraction, where input images traverse MobileNetV2's series of depth wise-separable convolutions. This architectural design substantially reduces computational complexity while maintaining high accuracy, making it ideal for large-scale image classification. Instead of traditional fully connected layers, Global Average Pooling (GAP) is applied to the feature maps, which reduces dimensionality and mitigates overfitting by summarizing spatial features across each channel. This is followed by a fully connected layer with 128 neurons activated by ReLU, refining the extracted features and enhancing the model's ability to differentiate between the four batik classes. A dropout layer is incorporated to further improve generalization by randomly deactivating neurons during training. The output layer utilizes a softmax activation function with four output nodes, corresponding to the classification labels: Kawung, Mega Mendung, Parang, and Truntum.

The model undergoes training for 50 epochs using the Adam optimizer, which efficiently adjusts learning rates to accelerate convergence [20]. The optimal learning rate is identified through the hyperparameter optimization process, ensuring stable and effective training. Early stopping is implemented to prevent overfitting by halting training if the validation loss does not improve for three consecutive epochs. This combination of architectural choices and training strategies enables the model to achieve high classification performance while maintaining computational efficiency, making it well-suited for practical batik classification applications.

2.5. Testing

After the training phase, the model is tested to evaluate its generalization performance on previously unseen images. The testing process follows the same preprocessing steps as training, including resizing each image to 128×128 pixels and normalizing it to maintain consistency in input representation. The preprocessed images are then fed into the MobileNetV2-based model [21], which extracts distinctive batik patterns and structural features. These extracted features are processed through fully connected layers, where the model assigns class probabilities to each of the four batik categories. The final classification is determined by selecting the class with the highest probability from the softmax layer. This testing phase assesses the model's capability to accurately differentiate between various batik motifs, ensuring it performs effectively on new, unseen data. By evaluating the model on a separate dataset, its robustness and reliability are validated, offering insights into its practical applicability for real-world batik classification tasks.

2.6. Evaluation

The model's performance is evaluated using a confusion matrix, which provides a detailed breakdown of classification results by comparing actual labels with predicted labels [22]. The confusion matrix is structured as a table where each row represents an actual class, and each column represents a predicted class. High values along the diagonal indicate correct classifications, while off-diagonal values highlight misclassifications. This visualization is crucial for identifying error patterns, such as which batik motifs are commonly confused with each other. Beyond the confusion matrix, various classification metrics are calculated to quantify the model's effectiveness [23]. Accuracy measures the overall proportion of correctly classified images, serving as a general performance indicator, formulated as equation (1).

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

Precision evaluates the proportion of correctly identified instances within each predicted class, ensuring that the model minimizes false positives, formulated as equation (2).

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

Recall, on the other hand, assesses the model's ability to correctly classify all instances of a given motif, which is critical for detecting underrepresented classes, formulated as equation (3).

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

The F1-score, as the harmonic mean of precision and recall, offers a balanced metric that accounts for both false positives and false negatives, formulated as equation (4).

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

By analyzing the confusion matrix alongside accuracy, precision, recall, and F1-score, a comprehensive understanding of the model's strengths and limitations in batik motif classification is obtained [24].

3. Result and Discussion

The performance of the batik classification model was evaluated using four hyperparameter optimization methods, namely Tree-structured Parzen Estimator (TPE), Gaussian Process Sampler (GPS), Grid Search, and Random Search. The comparison of their optimal hyperparameters and performance is summarized in Table 1.

From Table 1 it can be seen that TPE achieved the highest test accuracy of 91.94 percent, showing its effectiveness in finding near optimal hyperparameters. Both GPS and Grid Search produced the same test accuracy of 90.83 percent, although GPS is more efficient in the sampling process compared to the exhaustive nature of Grid Search.

Table 1. Performance Comparison of Sampling Algorithms.

Sampling Algorithm	Best Learning Rate	Best Drop Rate	Valid. Acc.	Test Acc.
TPE	0.00224	0.2583	92.36%	91.94%
GPS	0.00055	0.3483	92.01%	90.83%
Grid S.	0.00100	0.2000	92.71%	90.83%
Random	0.00031	0.4737	92.01%	88.61%

Random Search gave the lowest test accuracy at 88.61 percent, which indicates its lower reliability compared to more structured optimization methods. To provide a broader evaluation of classification performance, Table 2 reports additional metrics that include accuracy, precision, recall, and F1 score.

Table 2. Accuracy, Precision, Recall, and F1 Scores.

Sampling Algorithm	Accuracy	Precision	Recall	F1 Score
TPE	91.94%	92.44%	91.86%	92.09%
GPS	90.83%	90.81%	91.09%	90.89%
Grid S.	90.83%	90.87%	91.02%	90.87%
Random	88.61%	88.80%	88.74%	88.61%

Table 2 shows that TPE consistently outperformed the other optimization methods across all evaluation metrics. GPS and Grid Search had very similar results, both slightly below TPE, while Random Search again produced the weakest performance. These results confirm that structured optimization strategies, particularly TPE, are more effective in improving classification accuracy and the overall robustness of the batik classification model.

To provide a deeper insight into the classification performance, the detailed results are further illustrated using confusion matrices. The confusion matrix for the TPE method is presented in Figure 4, followed by those of the other optimization methods.

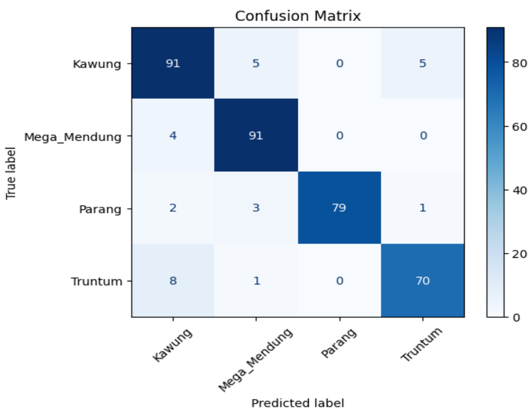

Figure 4. TPE sampling algorithm model accuracy and loss

TPE demonstrated strong performance, achieving a validation accuracy of 92.36% and the highest test accuracy of 91.94%. Its adaptive sampling strategy effectively balances exploration and exploitation, allowing it to focus on promising hyperparameter regions while refining selections based on previous evaluations. Compared to other methods, TPE outperformed GPSampler and Grid Search in test accuracy, indicating superior generalization to unseen data. This makes it particularly suitable for real-world applications where robustness across diverse inputs is critical. Meanwhile, the GP Sampling performance results can be seen in Figure 5.

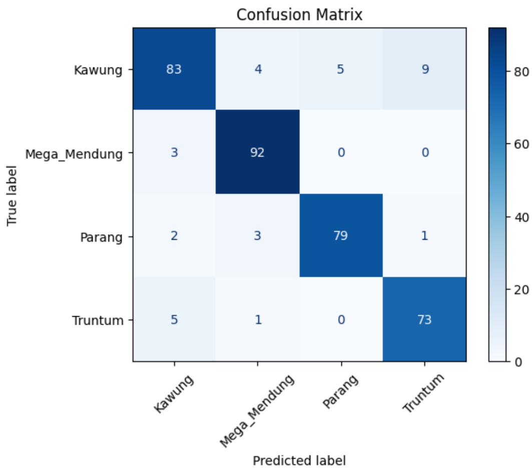

Figure 5. GP sampling algorithm confusion matrix.

GPSampler achieved a validation accuracy of 92.01% and a test accuracy of 90.83%, matching Grid Search in test performance while maintaining computational efficiency. By leveraging a probabilistic model, GPSampler efficiently navigates the hyperparameter space, making it a more resource-friendly alternative to exhaustive search techniques. Although its test accuracy was slightly lower than TPE's, it still demonstrated strong generalization capabilities. This suggests that GPSampler can be a viable choice when computational resources are limited but a structured optimization approach is still preferred.

Grid Search obtained the highest validation accuracy at 92.71%, indicating that systematically evaluating all possible hyperparameter combinations within a predefined space can lead to strong performance. However, its test accuracy (90.83%) was slightly lower than that of TPE, suggesting that it may overfit the training data despite its thorough search process. Additionally, Grid Search is computationally expensive, making it impractical for large-scale optimization tasks. While it is effective when the search space is well-defined and small, TPE's more adaptive approach proved more efficient in identifying configurations that generalize better. Result of Grid Search performance can be seen in Figure 6.

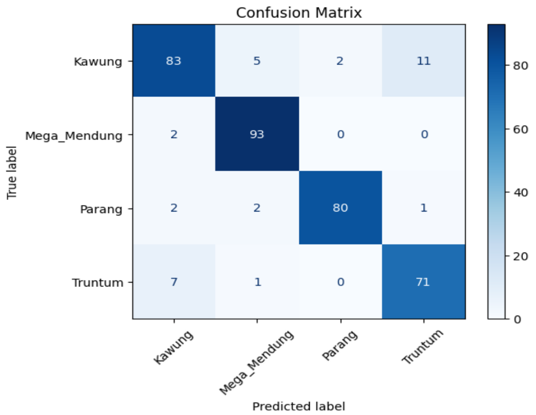

Figure 6. Grid Search sampling algorithm confusion matrix.

Among all methods, Random Search, as seen in Figure 7 yielded the lowest validation accuracy (92.01%) and test accuracy (88.61%). Since it selects hyperparameters randomly, it lacks the structured approach of Bayesian optimizers and Grid Search, often leading to suboptimal configurations. While Random Search is computationally inexpensive and capable of covering a broad range of hyperparameters, its lower accuracy highlights its inefficiency in finding the best-performing model. Compared to TPE, GPSampler, and Grid Search, Random Search is the least reliable for optimizing deep learning models, especially when higher accuracy and generalization are priorities.

These results highlight the advantages of Bayesian optimization, particularly TPE, which strikes a balance between search efficiency and model performance. While Grid Search attained the highest validation accuracy, TPE's superior generalization capability makes it a more suitable choice for scenarios where computational efficiency and adaptability to unseen data are crucial.

The comparison of test accuracy across different sampling methods is illustrated in Fig. 12., which visually represents the performance differences among the approaches discussed.

The test accuracy results indicate notable variations among different sampling algorithms, reflecting their effectiveness in hyperparameter optimization. The highest test accuracy was achieved using the TPE method (91.94%), suggesting its superior capability in identifying an optimal hyperparameter configuration. This aligns with previous findings that highlight TPE's efficiency in exploring the hyperparameter space adaptively, allowing for better convergence to optimal values.

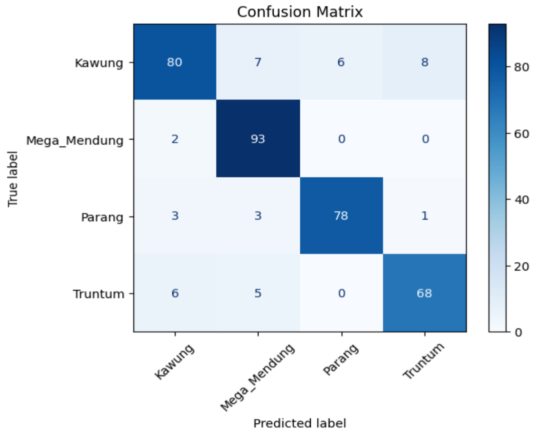

Figure 7. Random sampling algorithm confusion matrix.

Conversely, both GPSampler and Grid Search yielded identical test accuracies of 90.83%. While Grid Search systematically evaluates all possible combinations within a predefined search space, its exhaustive nature may lead to suboptimal efficiency. GPSampler, a Gaussian process-based method, also demonstrated comparable performance, suggesting its potential in hyperparameter selection, although it did not surpass TPE in this case.

The lowest test accuracy was observed with the Random search method (88.61%). Given its purely stochastic nature, this outcome is expected, as random sampling does not leverage prior knowledge or adaptive strategies to refine the search process. The performance gap between Random search and TPE (3.33%) underscores the advantage of more structured search strategies in achieving higher accuracy.

Overall, the results demonstrate that sampling strategies significantly impact model performance, with TPE emerging as the most effective in maximizing test accuracy. As shown in Figure 8, the bar chart comparison highlights that TPE outperforms the other methods.

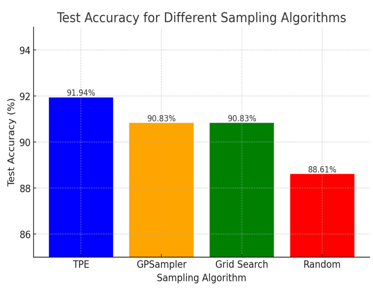

Figure 8. Comparison of test accuracy across sampling methods

Future work could further explore hybrid approaches or ensemble-based optimization techniques to enhance model generalization.

4. Conclusion

This study examined the effectiveness of various hyperparameter optimization sampling algorithms, namely TPE, GPSampler, Grid Search, and Random Search, in enhancing the classification accuracy of a MobileNetV2-based batik pattern recognition model. The findings revealed that TPE achieved the highest test accuracy of 91.94%, showcasing its superior capability to balance exploration and exploitation in hyperparameter tuning. Although Grid Search attained the highest validation accuracy (92.71%), its performance on unseen test data was slightly lower, indicating that an exhaustive search does not always lead to better generalization. GPSampler also delivered competitive results, achieving 90.83% test accuracy while being more computationally efficient than Grid Search. In contrast, Random Search resulted in the lowest test accuracy (88.61%), highlighting its inefficiency in identifying optimal configurations. These results suggest that Bayesian optimization techniques, especially TPE, are more effective in tuning deep learning models for batik classification, offering a favorable trade-off between computational cost and model generalization. Future research could investigate additional hyperparameter search strategies, explore model architectures beyond MobileNetV2, and utilize more diverse batik datasets to further enhance classification performance. Additionally, incorporating explainability techniques could improve interpretability, making AI-driven batik recognition more accessible for cultural heritage preservation and practical applications.

References

[1] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for seg menting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289.

[2] Anggoro, D. A., Marzuki, A. A. T., and Supriyanti, W. 2023. Classification of Solo Batik patterns using deep learning convolutional neural networks algorithm. TELKOMNIKA (Telecommunication Computing Electronics and Control), 22(1), 232. https://doi.org/10.12928/telkomnika.v22i1.24598

[3] Auliaddina, S., and Arifin, T. 2024. Penggunaan Data Augmentasi dan Hyperparameter Tuning dalam Klasifikasi Jenis Batik menggunakan Model CNN. SISTEMASI: Jurnal Sistem Informasi, 13(1).

[4] Azhar, Y., and Akbi, D. R. 2024. Performance Comparison of GLCM Features and Preprocessing Effect on Batik Image Retrieval. JOIV: International Journal on Informatics Visualization, 8(3), 1339. https://doi.org/10.62527/joiv.8.3.2179

[5] Roland, R., Angelica, C., Diputra, J. A., Azizul, Z. H., and Fitrianah, D. 2024. CNN Classifier Parameter Optimization with Genetic Algorithms: A Case Study of Indonesian Batik Patterns. International Journal of Computational Intelligence and Applications, 23(02). https://doi.org/10.1142/S1469026824500044

[6] Murinto, M., Winiarti, S., and Faisal, I. 2024. Particle Swarm Optimization Algorithm for Hyperparameter Convolutional Neural Network and Transfer Learning VGG16 Model. Journal of Computer Science, Information Technology and Telecommunication Engineering. https://doi.org/10.30596/jcositte.v5i1.16680

[7] Vo, H.-T., Ngoc, H. T., and Quach, L.-D. 2023. An Approach to Hyperparameter Tuning in Transfer Learning for Driver Drowsiness Detection Based on Bayesian Optimization and Random Search. International Journal of Advanced Computer Science and Applications, 14(4). https://doi.org/10.14569/IJACSA.2023.0140492

[8] Elvitaria, L., Shaubari, E. F. A., Samsudin, N. A., Khalid, S. K. A., S., -, and Indra, Z. 2024. A Proposed Batik Automatic Classification System Based on Ensemble Deep Learning and GLCM Feature Extraction Method. International Journal of Advanced Computer Science and Applications, 15(10). https://doi.org/10.14569/IJACSA.2024.0151058

[9] Yong, L., Ma, L., Sun, D., and Du, L. 2023. Application of MobileNetV2 to waste classification. PLOS ONE, 18(3), e0282336. https://doi.org/10.1371/journal.pone.0282336

[10] Meranggi, D. G. T., Yudistira, N., and Sari, Y. A. 2022. Batik Classification Using Convolutional Neural Network with Data Improvements. JOIV: International Journal on Informatics Visualization, 6(1), 6. https://doi.org/10.30630/joiv.6.1.716

[11] Nurhaida, I., Ayumi, V., Fitrianah, D., Zen, R. A. M., Noprisson, H., and Wei, H. 2020. Implementation of deep neural networks (DNN) with batch normalization for batik pattern recognition. International Journal of Electrical and Computer Engineering (IJECE), 10(2), 2045. https://doi.org/10.11591/ijece.v10i2.pp2045-2053

[12] Khaled, A. 2025. BCN: Batch channel normalization for image classification. In International Conference on Pattern Recognition, 295–308. Springer, Cham.

[13] Tang, D., Yang, N., Deng, Y., Zhang, Y., Sani, A. S., and Yuan, D. 2025. Stability-driven CNN training with Lyapunov-based dynamic learning rate. In Australasian Database Conference, 58–70. Springer, Singapore.

[14] Mandala, S., Jatmiko, W., Nurmaini, S., and Rizal, A. 2025. OCADN: Improving Accuracy in Multi-Class Arrhythmia Detection From ECG Signals With a Hyperparameter-Optimized CNN. IEEE Access.

[15] Hussain, W., Mushtaq, M. F., Shahroz, M., Akram, U., Ghith, E. S., Tlija, M., and Ashraf, I. 2025. Ensemble genetic and CNN model-based image classification by enhancing hyperparameter tuning. Scientific Reports, 15(1), 1003.

[16] Legashev, L., Tolmachev, S., Bolodurina, I., Shukhman, A., and Grishina, L. 2024. Investigation into the Hyperparameters of Error-Based Adaptive Sampling Approach for Surrogate Modeling. Modelling, 5(4), 2051–2074. https://doi.org/10.3390/modelling5040106

[17] Kurniawan, K., Windarto, A. P., and Solikhun, S. 2025. Refining CNN architecture for forest fire detection:

improving accuracy through efficient hyperparameter tuning. Bulletin of Electrical Engineering and Informatics, 14(2), 1202–1211.

[18] Li, J., Huang, Z., Jiang, L., and Zhang, Y. 2025. An intelligent fault diagnosis model for bearings with adaptive hyperparameter tuning in multi-condition and limited sample scenarios. Scientific Reports, 15(1), 10095.

[19] Fan, Z., Sohail, S., Sabrina, F., and Gu, X. 2024. Sampling-Based Machine Learning Models for Intrusion Detection in Imbalanced Dataset. Electronics, 13(10), 1878. https://doi.org/10.3390/electronics13101878

[20] Z. Zhang, "Improved adam optimizer for deep neural networks," in Proceedings of the IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Jun. 2018, pp. 1–2.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.

[22] S. Suyahman and A. Hapsari, "VGG-based feature extraction for classifying traditional batik motifs using machine learning models," Preservation, Digital Technology & Culture, 2025.

[23] A. Wicaksono, D. Prasetyo, Y. Mar'atullatifah, D. U. Iswavigra, H. Mahmudah, and A. Hapsari, "Data analysis and explainable machine learning for stunting prediction," Journal of Artificial Intelligence and Legal Technology, vol. 1, no. 1, pp. 35–44, 2025.

[24] A. Hapsari, "Optimized machine learning with TPE for air quality classification and public health risk estimation," Journal of Artificial Intelligence and Legal Technology, vol. 1, no. 1, pp. 9–14, 2025.