A mobile-optimized convolutional neural network approach for real-time batik pattern recognition

Rosalina[1], Genta Sahuri[2], Hana Desriva[1]
[1]Department of Informatics, Faculty of Computing, President University, Bekasi, Indonesia
[2]Department of Information System, Faculty of Computing, President University, Bekasi, Indonesia

Article Info

Article history:

Received Jan 20, 2024
Revised Feb 16, 2024
Accepted Mar 5, 2024

Keywords:

Batik
Convolutional neural network
Geometric pattern
Mobile application
Pattern recognition
Real-time

ABSTRACT

This research focuses on preserving and sharing knowledge about Indonesian batik, a blend of art and technology symbolizing the nation's creativity. To address declining awareness of batik types, a mobile application is introduced for real-time recognition and classification of batik motifs. The goal is to maintain appreciation and understanding of this cultural heritage. Using the EfficientNet convolutional neural network (CNN) architecture, the study enhances model accuracy with effective scaling. A dataset of 1350 images representing 15 batik types supports robust model training and evaluation. Results demonstrate successful implementation, yielding an Android app capable of deep learning-based real-time recognition with an 83% accuracy rate. This innovation aims to empower users to identify and appreciate distinct batik types, ensuring cultural preservation for current and future generations.

Corresponding Author:

Rosalina
Department of Informatics, Faculty of Computing, President University
Bekasi, Indonesia
Email: rosalina@president.ac.id

1. INTRODUCTION

The art of batik in Indonesia flourishes with a kaleidoscope of patterns, each originating uniquely from different regions. Craftsmen and artists contribute to the creation of thousands of distinct batik patterns (depicted in Figure 1), forming a cultural tapestry that is both intricate and expansive. Geometric patterns exhibit mathematical precision, while non-geometric ones evoke a more fluid and natural aesthetic. The significance of these patterns lies in their capacity to characterize and represent specific types of batik. Typically arranged skillfully in repetitive sequences, these patterns create a visually captivating and cohesive narrative across the fabric. Through their arrangement, the fundamental essence of each pattern is artistically expressed, contributing significantly to the broader cultural significance of Indonesian batik. Despite the overarching similarities observable across various batik patterns, subtle differentiations persist in finer details, such as color utilization. It is within these nuanced variations that the true diversity and richness of batik patterns emerge. Each class embodies a distinctive visual language, perpetuating a cultural, and historical narrative expressed through the artistry of batik patterns. This preservation of cultural identity and history continues to be celebrated and communicated through the intricate and meaningful designs of Indonesian batik.

In batik pattern recognition, previous research has laid a foundation for understanding the complexities inherent in this art form. While these studies have made significant strides, there are notable gaps that remain to be addressed. Specifically, earlier investigations have primarily focused on the technical aspects of pattern recognition, such as algorithm development and accuracy metrics. However, they have not explicitly delved into the broader cultural significance of batik patterns and their impact on preserving Indonesia's cultural heritage. For instance, while earlier studies have explored the effectiveness of algorithms like the features from accelerated segment test (FAST) corner detection algorithm, K-nearest neighbor (KNN), support vector machine (SVM), and deep neural networks (DNNs) in identifying batik patterns [1]–[10], they have not explicitly addressed how these technological advancements contribute to the broader goals of cultural preservation and education. The focus has largely been on achieving high-precision recognition without necessarily considering the implications for cultural heritage. Additionally, a study implemented a DNN [11], [12] indicating the broad spectrum of cutting-edge technologies employed in the pursuit of effective batik pattern recognition. A pivotal contribution to this burgeoning field comes from the research conducted by [13]–[25] which zeroes in on the application of convolutional neural networks (CNNs) for the identification of six distinct batik patterns. The experimental results study [13] are particularly noteworthy, revealing impressive performance metrics. The reported accuracy of 94% and a top-2 accuracy of 99% attest to the efficacy of the CNN approach in achieving high-precision batik pattern recognition. As a cornerstone of deep learning, CNNs have demonstrated remarkable capabilities in image recognition tasks, making them well-suited for the nuanced and intricate patterns found in batik. This research draws inspiration from previous studies and advancements in the field of image recognition, particularly those focusing on cultural heritage preservation and mobile optimization.

Figure 1. Indonesian batik pattern

2. METHOD

The methodology for time batik pattern recognition involves five crucial stages, depicted in Figure 2: data collection, data preprocessing, model selection, ablation study, and result analysis. Firstly, relevant data on batik patterns is collected systematically. Then, the data undergoes preprocessing to ensure quality and consistency. Next, appropriate models are selected for training and testing. Following this, an ablation study is conducted to assess the impact of different components on model performance. Finally, results are analyzed comprehensively to evaluate the effectiveness and efficiency of the proposed approach in recognizing batik patterns in real-time. Figure 2 provides a visual representation of the methodology, depicting the flow of the five stages involved in time batik pattern recognition. This structured approach ensures that the research is conducted systematically, allowing for the replication of experiments and results by other researchers in the field.

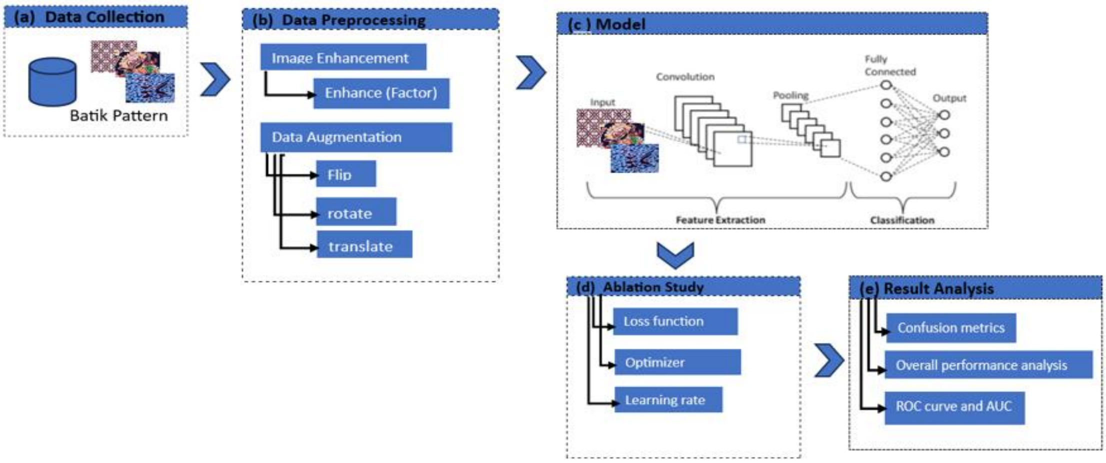

Figure 2. Methodology for batik pattern recognition

2.1. Data collection

The datasets utilized in this research were sourced from Google Images, representing a collection that was subsequently augmented to address limitations in the initially obtained images. The resulting batik dataset comprises a total of 1315 images categorized into 15 distinct classes, each representing a different type of batik motif. The 15 types of batik are: Batik Bali, Batik Betawi, Batik Cendrawasih, Batik Dayak, Batik Geblek Renteng, Batik Ikat Celup, Batik Insang, Batik Kawung, Batik Lasem, Batik Megamendung, Batik Pala, Batik Parang, Batik Poleng, Batik Sekar Jagad, and Batik Tambal.

These motifs exhibit significant variations in their designs, presenting both an interesting and challenging dataset for image classification tasks. Each image in the dataset adheres to a standardized size of 224×224 pixels, a common format employed in image classification tasks. This particular size is especially suitable for CNNs, which often necessitate fixed input sizes. The consistent dimensions across the dataset facilitate the development and application of robust models for the recognition and classification of diverse Indonesian batik motifs. Researchers can leverage this dataset to delve into the complexities of batik art and advance image classification techniques within the context of cultural preservation. The Batik Kawung dataset before augmentation is shown in Figure 3, while the augmented version is depicted in Figure 4.

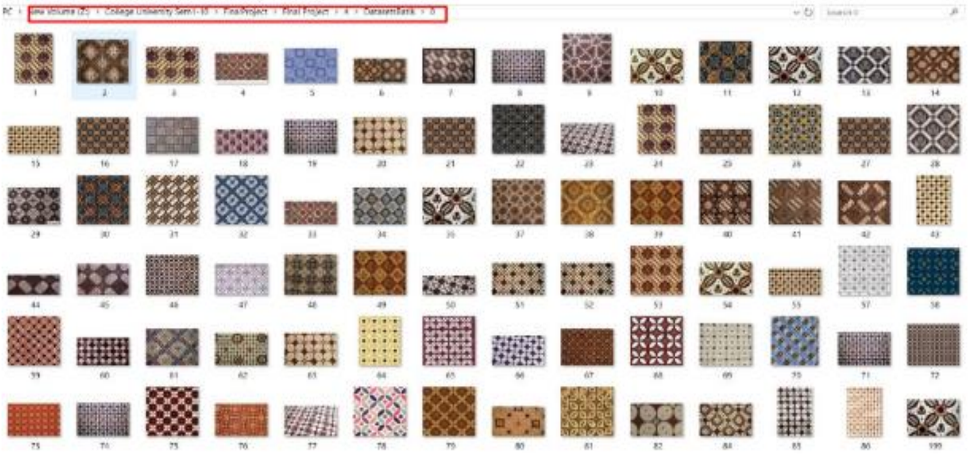
PC › New Volume (Z:) › College University Sem1-10 › FinalProject › Final Project › 4 › DatasetsBatik › 0

Figure 3. Batik Kawung dataset before augmentation

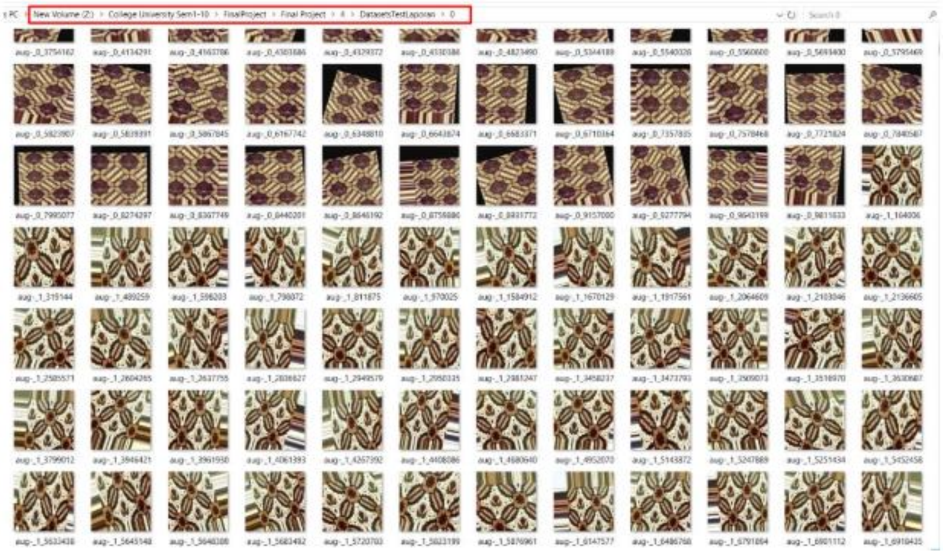
PC › New Volume (Z:) › College University Sem1-10 › FinalProject › Final Project › 4 › DatasetsTestLaporan › 0

Figure 4. Batik Kawung dataset after augmentation

2.2. Data preprocessing

In batik pattern recognition, data preprocessing is essential to optimize the dataset for accurate model training and classification. Initially, resizing images to a standardized format, often 224×224 pixels, ensures uniformity and compatibility across the dataset. This uniformity facilitates consistent processing by subsequent stages of the recognition pipeline, minimizing computational complexities and improving efficiency. Normalizing pixel values is equally critical as it standardizes the intensity levels across all images, reducing variability caused by differences in lighting conditions or camera settings. Additionally, handling outliers, such as images with excessive noise or irrelevant content, is crucial for maintaining dataset integrity and preventing biases during model training. By systematically addressing these preprocessing tasks, the dataset's quality and consistency are enhanced, leading to improved model performance and robustness in identifying and classifying batik patterns. Ultimately, this meticulous preprocessing ensures that the batik pattern recognition system can effectively preserve cultural heritage by accurately capturing the intricacies and nuances of Indonesian batik art across diverse motifs and designs.

2.3. Model selection

In the process of model selection for batik pattern recognition, various factors are carefully considered to ensure optimal performance and efficiency. Firstly, the complexity of the dataset is assessed, taking into account the intricacies and variations present in batik motifs. Models must be capable of capturing the diverse patterns and textures characteristic of batik art while maintaining high accuracy in classification. Additionally, available computational resources play a crucial role in determining the feasibility of deploying certain models. Models that require extensive computational power or memory may not be practical for implementation, especially in real-time recognition systems. Among the plethora of available options, CNNs stand out as a popular choice for batik pattern recognition. CNNs are specifically designed to handle image data effectively, making them well-suited for tasks such as identifying and classifying batik motifs. The architecture of a typical CNN consists of convolutional layers followed by pooling layers, which extract hierarchical features from input images. These features are then fed into fully connected layers for classification. The formula for a basic convolutional layer in a CNN can be expressed as in (1):

$$H_{i,j}^{l} = f\left(\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} W_{m,n}^{l} . X_{i+m,j+n}^{l-1} + b^{l}\right) \tag{1}$$

Where $H_{i,j}^{l}$ is the output of the convolutional layer at position (i, j) in the feature map l, $W_{m,n}^{l}$ represents the weights of the convolutional filter at position (m, n), $X_{i+m,j+n}^{l-1}$ denotes the input feature map from the previous layer, and b^{l} is the bias term. The function f represents the activation function, such as rectified linear unit (ReLU), applied element-wise to the convolutional output.

2.4. Ablation study

In batik pattern recognition, conducting an ablation study is crucial for dissecting the performance of the selected model and understanding the contribution of its individual components. This study involves systematically modifying or removing specific components of the model to evaluate their impact on recognition accuracy. By isolating and manipulating various elements within the model architecture, researchers can gain insights into which features or modules play a significant role in enhancing performance. The ablation study typically begins by identifying key components within the model, such as specific layers or techniques, that are believed to influence recognition accuracy. These components may include convolutional layers, pooling layers, activation functions, regularization techniques, or optimization algorithms. Once identified, these components are systematically altered or excluded from the model architecture. During the ablation study, the modified model configurations are trained and evaluated using the same dataset used in the initial model training. The performance metrics, such as accuracy, precision, recall, and F1-score, are then compared with those obtained from the original model. This comparison allows researchers to quantify the impact of each modified component on the model's overall performance. The formula for conducting an ablation study can be expressed as (2):

$$\Delta Performance = Performance_{modified} - Performance_{original} \tag{2}$$

Where $\Delta Performance$ represents the change in performance resulting from modifying or removing specific components, $Performance_{modified}$ denotes the performance metrics obtained from the modified model configuration and Performance$_{\text{original}}$, and $Performance_{original}$ denotes the performance metrics obtained from the original model configuration.

2.5. Result analysis

In result analysis step the performance of the model is rigorously evaluated to assess its effectiveness and efficiency. This evaluation encompasses various metrics such as accuracy, precision, recall, and F1-score, providing a comprehensive understanding of the model's capabilities in correctly identifying and classifying batik patterns. These metrics offer insights into different aspects of the model's performance, including its ability to minimize misclassifications and accurately distinguish between different batik motifs. The formula for these performance metrics is as follows:

– Accuracy measures the proportion of correctly classified instances among the total number of instances:

$$Accuracy = \frac{Number\ of\ Correct\ Prediction}{Total\ Number\ of\ Prediction} \tag{3}$$

– Precision quantifies the proportion of true positive predictions among all positive predictions made by the model:

$$Precision = \frac{True\ Positives}{True\ Positive+False\ Positive} \tag{4}$$

– Recall calculates the proportion of true positive predictions among all actual positive instances in the dataset:

$$Recall = \frac{True\ Positives}{True\ Positive+False\ Negative} \tag{5}$$

Additionally, the effectiveness and efficiency of the proposed approach in recognizing batik patterns in real-time are evaluated based on the analysis of results, considering factors such as processing speed and resource utilization.

3. RESULTS AND DISCUSSION

This study explored the effects of various configurations on model performance in batik pattern recognition. While previous research has investigated similar topics, they often lacked explicit analysis of the influence of specific parameters on model outcomes. Our analysis revealed significant shortcomings in the initial model configuration, with accuracy plateauing at 26% and a remarkably high loss of approximately 158%. Despite adjustments to parameters such as learning rate and batch size, suboptimal performance persisted. Higher epochs led to improved accuracy but resulted in overfitting, highlighting the importance of balancing model complexity and training duration. Furthermore, modifications in batch size and filter sizes yielded varying outcomes, underscoring the need for careful parameter tuning. While our study provided valuable insights into model performance under different configurations, further investigation is necessary to address persistent challenges such as overfitting. Future research should focus on refining model architectures and exploring alternative optimization strategies. Our findings underscore the iterative nature of model development and the importance of continuous experimentation. Future studies may benefit from exploring novel regularization techniques and evaluating model performance across diverse datasets. Despite numerous attempts to optimize model performance, challenges such as overfitting persist. The iterative experimentation process has provided valuable insights into the impact of various configurations on model outcomes. Continued refinement and exploration of alternative strategies are essential for developing a robust and generalizable model for batik pattern recognition.

The model training experiment, employing the Adam optimizer with a learning rate of 0.001, a batch size of 8, and 40 epochs, yielded suboptimal results. Despite a dataset comprising 400 samples split into 8 batches per epoch, the model, comprising three convolutional layers, failed to achieve satisfactory performance, as shown in Figure 5. The initial layer, featuring 60 filters with a 5×5 filter size, followed by the 2nd and 3rd layers with 30 filters each and a 3×3 filter size, inadequately captured underlying patterns in the data. Consequently, the trained model exhibited significant shortcomings, with accuracy plateauing at 26% and a remarkably high loss of approximately 158%. These outcomes underscore the necessity for a re-evaluation of the model architecture, hyperparameters, or dataset quality to bolster the model's learning capabilities and attain more favorable results in subsequent training iterations.

Despite adjusting the learning rate to 0.0001, the model's performance remains unsatisfactory, with the loss persisting at a high level near 140%. This modification fails to substantially enhance accuracy, which only marginally improves to 40%. The persistence of high loss suggests that the model struggles to effectively learn from the data, indicating underlying issues with its architecture or hyperparameters. Further adjustments or re-evaluation may be necessary to address these shortcomings and improve the model's ability to capture patterns in the dataset, ultimately leading to better performance in subsequent training iterations as depicted in Figure 6.

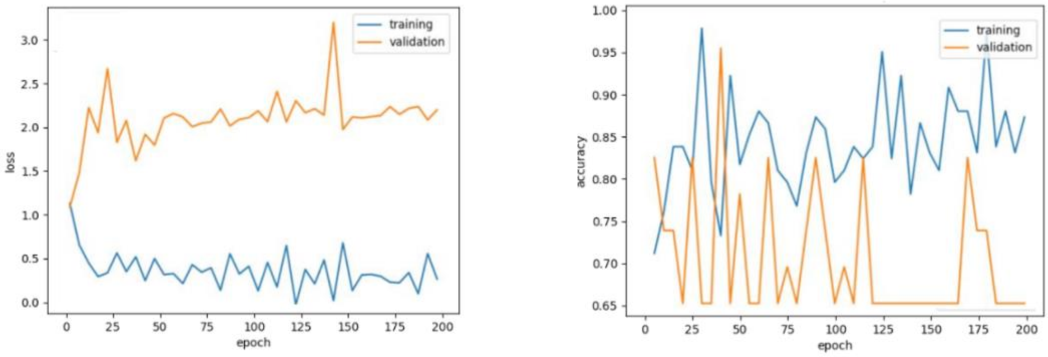

Figure 5. Testing model learning rate 0.001, batch size 8.40 epochs

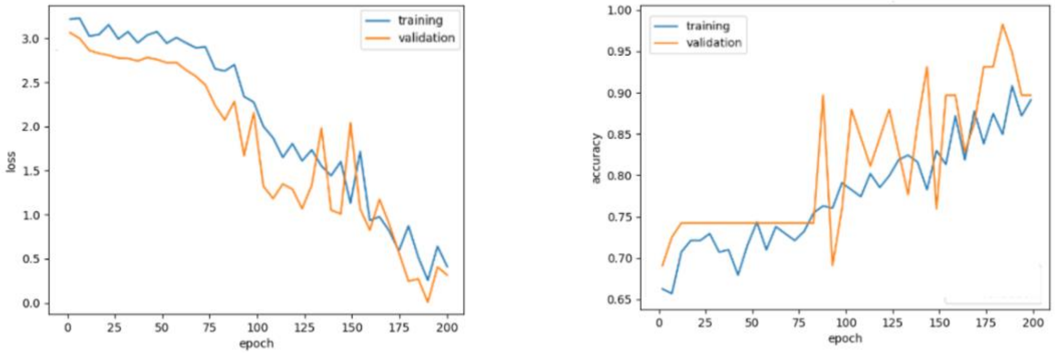

Figure 6. Testing model change the learning rate to 0.0001

Changing the batch size to 2 has resulted in a noticeable discrepancy between the training and validation metrics. While the training loss approaches 0% and accuracy reaches nearly 100%, the validation loss spikes to 300%, with accuracy dropping to around 60%. This stark contrast signifies overfitting, where the model memorizes training data excessively but fails to generalize to new data. Figure 7 visually demonstrates this overfitting phenomenon. To mitigate it, strategies such as reassessing model complexity, incorporating regularization techniques, or implementing early stopping are recommended. These approaches aim to promote better generalization and prevent the model from becoming too specialized to the training data. Modifying the number of filters in the convolutional layers from 60 to 128 and 30 to 256 has maintained robust training performance, akin to the previous setup. However, the validation loss has soared to almost 600%, signaling severe overfitting. Despite low training loss, the model struggles to generalize, evident in the significant performance gap between training and validation datasets. To address this, revisiting the model architecture, exploring regularization techniques, or implementing dropout may help enhance generalization. Figure 8 visually emphasizes the clear overfitting trend observed in the results, underscoring the need for interventions to ensure the model's effectiveness across diverse datasets.

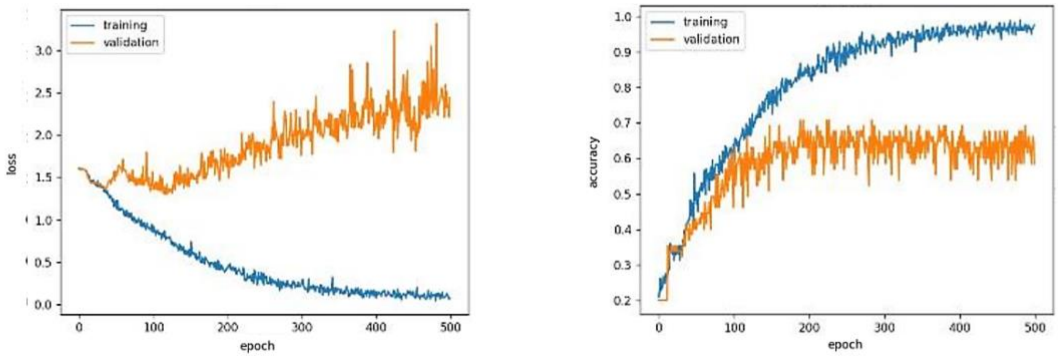

Figure 7. Testing model change the batch size to 2

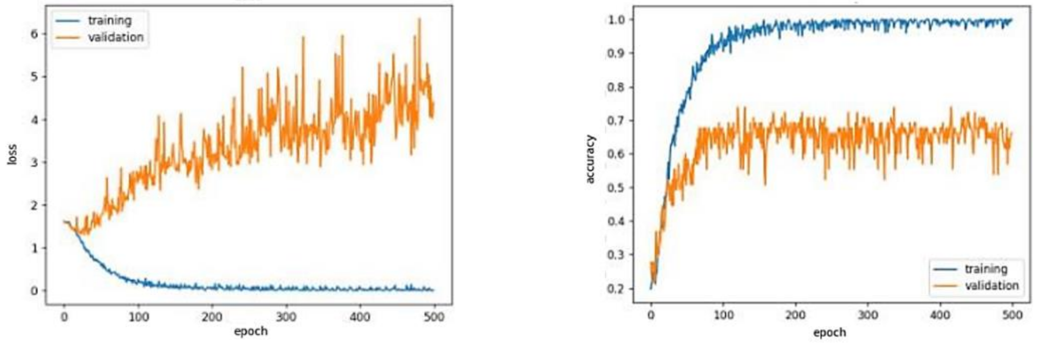

Figure 8. Testing model change the convolutional layer filter from 60 to 128, and 30 to 256

In response to the persistent overfitting issue, the dropout technique has been introduced to the model. Dropout involves randomly deactivating neurons during training, preventing over-reliance on specific pathways and enhancing generalization. With a dropout rate of 0.25 applied to each layer, the model's performance has been reevaluated. Figure 9 illustrates the outcomes, showcasing the impact of dropout on the training and validation metrics. This technique introduces a degree of randomness during training, helping to mitigate overfitting by promoting a more robust and generalized representation of the data. Careful consideration of dropout rates and its application across different layers is essential for achieving a balanced and effective regularization effect.

Adjusting the filter sizes in the convolutional layers, specifically reducing the first layer's size to 32 and the subsequent layers to 64, has been implemented in pursuit of a more effective model. The obtained results, depicted in Figure 10, reveal the impact of this modification on the training and validation metrics. Fine-tuning filter sizes is a crucial aspect of neural network design, influencing the model's ability to capture relevant features. By iteratively experimenting with configurations and monitoring performance, a balance can be struck to achieve improved accuracy and prevent overfitting. The visual representation in Figure 10 provides insights into how this adjustment influences the model's behavior across training epochs.

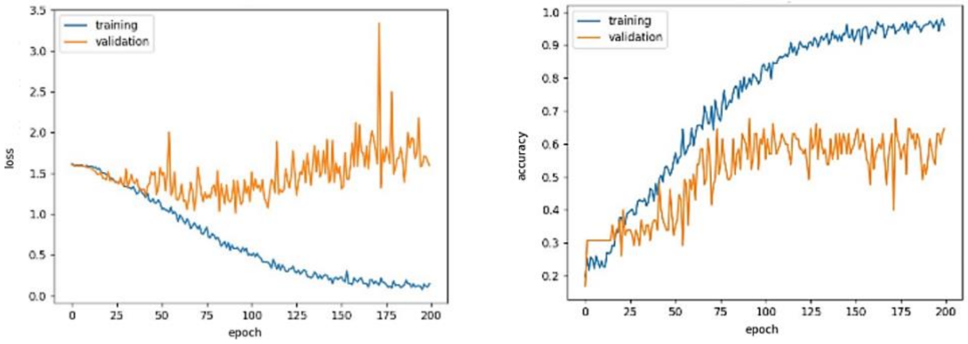

Figure 9. Testing model add dropout 0.25 for each layer

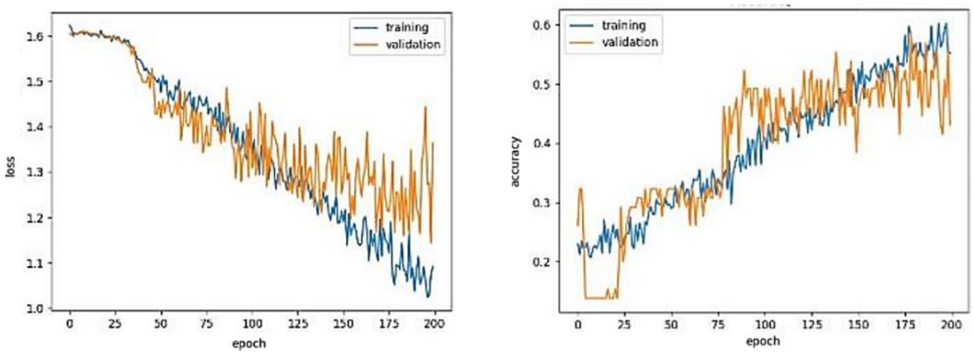

Figure 10. Testing model change the convolutional layer filter from 128 to 32, and 256 to 64

Increasing the dataset size through augmentation is a positive step to address overfitting. The iterative adjustments to hyperparameters, specifically changing the learning rate from 0.0005 to 0.001 and then to 0.0005, along with variations in batch size and epochs, have led to improved accuracy. However, the persistence of a high validation loss, around 40%, suggests that further optimization may be needed. Consider experimenting with different augmentation techniques, adjusting regularization methods, or fine-tuning the model architecture to enhance generalization. Achieving a balance between model complexity and dataset augmentation is crucial. The depicted outcomes in Figure 11 showcase promising accuracy, but a focus on reducing validation loss remains vital for ensuring the model's effectiveness across diverse datasets. Continued experimentation and careful parameter tuning are essential for refining the model's performance. It's promising to see that modifying the model architecture by adding more layers and adjusting filter sizes has led to significant improvements. The use of 4 CNN layers with varying numbers of filters (32, 32, 64, 128) and a switch to 3×3 filters has resulted in remarkable outcomes.

Increasing the number of epochs from 300 to 1,000 has further improved the accuracy and loss, as depicted in Figure 12. Transitioning from grayscale (1 channel) to RGB (3 channels) has also been implemented to capture richer color information. Despite occasional fluctuations in the figures, the overall precision and recall values during predictions have significantly improved. This highlights the iterative nature of model development, where numerous attempts and adjustments are needed to achieve the best results. The incorporation of RGB channels often enhances the model's ability to discern intricate patterns and features in the data. Continued experimentation, monitoring, and refinement are key aspects of optimizing deep learning models for specific tasks. It's commendable that, after 60 attempts, a model with good precision and recall values has been achieved.

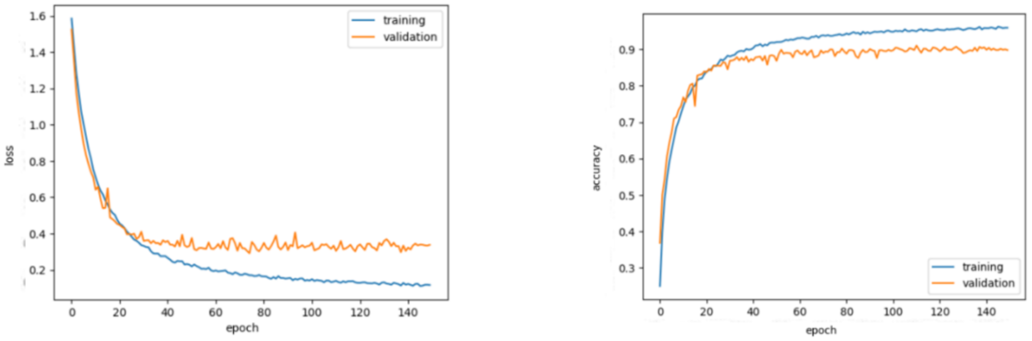

Figure 11. Testing change learning rate to 0.0005, with batch size 32, epoch 150

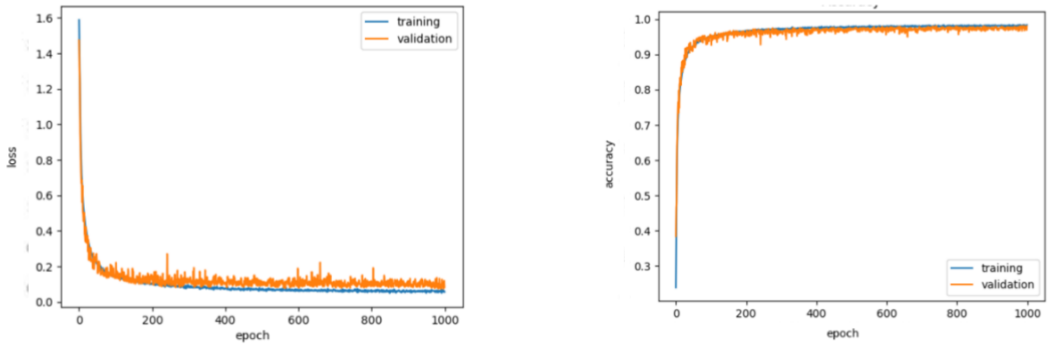

Figure 12. Testing with 1,000 epoch (RGB channel)

When testing applications utilizing the TensorFlow lite (TFLite) model, it is crucial to evaluate the model's performance on the Android platform, focusing on accuracy and its capability to handle diverse datasets. This includes assessing the model's ability to correctly classify images and generate pertinent outputs based on input data. Furthermore, it is essential to examine the model's robustness under various conditions, such as changing lighting and camera angles, to ensure consistent performance. Additionally, one must consider the computational resources needed by the TFLite model and the Android device's capacity to meet these requirements. Thorough testing of the TFLite model within the Android application is essential to guarantee that the application operates as intended and meets user expectations. The outcomes are illustrated in Figure 13.

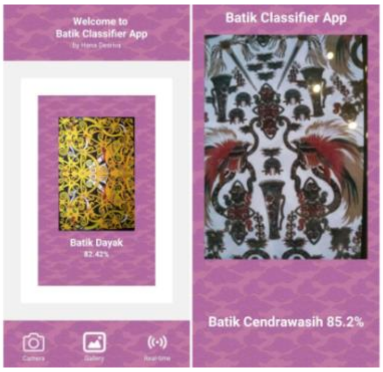

Figure 13. The result of testing applications utilizing the TFLite model

4. CONCLUSION

This research addresses the imperative of preserving Indonesian batik, a cultural amalgamation of art and technology reflecting the nation's creativity. Faced with diminishing contemporary awareness of diverse batik types, the study introduces a pioneering solution—a mobile application utilizing real-time recognition and classification of batik motifs and characteristics. By employing the EfficientNet architecture of CNN, the model achieves heightened accuracy through effective scaling. Implemented in Python and seamlessly integrated with Android through TFLite, the research yields a user-friendly mobile application capable of deep learning-based real-time recognition, boasting an impressive accuracy of 83%. The comprehensive dataset, encompassing 1350 images representing 15 unique batik types, ensures a robust foundation for model training and evaluation. The batik classifier not only demonstrates the efficacy of CNN in image recognition but also highlights the utilization of TensorFlow Lite for efficient on-device processing, emphasizing the synergy of tradition and technology. This application empowers users to discern and appreciate diverse batik types, fostering cultural preservation for present and future generations. Ultimately, the research underscores the successful convergence of cultural heritage and technological innovation in the realm of batik appreciation.

REFERENCES

[1] B. J. Filia *et al.*, "Improving batik pattern classification using CNN with advanced augmentation and oversampling on imbalanced dataset," *Procedia Computer Science*, vol. 227, pp. 508–517, 2023, doi: 10.1016/j.procs.2023.10.552.

[2] A. E. Minarno, M. C. Mustaqim, Y. Azhar, W. A. Kusuma, and Y. Munarko, "Deep convolutional generative adversarial network application in batik pattern generator," in *2021 9th International Conference on Information and Communication Technology (ICoICT)*, IEEE, Aug. 2021, pp. 54–59, doi: 10.1109/ICoICT52021.2021.9527514.

[3] A. F. Ihsan, "Initial study of batik generation using variational autoencoder," *Procedia Computer Science*, vol. 227, pp. 785–794, 2023, doi: 10.1016/j.procs.2023.10.584.

[4] A. D. Wibawa, E. A.Wicaksono, S. D. Suryani, and R. Rumadi, "Javanese batik image classification using self-organizing map," in *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, IEEE, Feb. 2023, pp. 472–477, doi: 10.1109/ICCoSITE57641.2023.10127783.

[5] A. H. Rangkuti, A. Harjoko, and A. Putra, "A novel reliable approach for image batik classification that invariant with scale and rotation using MU2ECS-LBP algorithm," *Procedia Computer Science*, vol. 179, pp. 863–870, 2021, doi: 10.1016/j.procs.2021.01.075.

[6] A. E. Minarno, Y. Azhar, F. D. S. Sumadi, and Y. Munarko, "A robust batik image classification using multi texton co-occurrence descriptor and support vector machine," in *2020 3rd International Conference on Intelligent Autonomous Systems (ICoIAS)*, IEEE, Feb. 2020, pp. 51–55, doi: 10.1109/ICoIAS49312.2020.9081833.

[7] B. S. Negara, E. Satria, S. Sanjaya, and D. R. D. Santoso, "ResNet-50 for classifying Indonesian batik with data augmentation," in *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, 2021, pp. 1–4, doi: 10.1109/ICOTEN52080.2021.9493488.

[8] E. Winarno, W. Hadikurniawati, A. Septiarini, and H. Hamdani, "Analysis of color features performance using support vector machine with multi-kernel for batik classification," *International Journal of Advances in Intelligent Informatics*, vol. 8, no. 2, pp. 151-164, 2022, doi: 10.26555/ijain.v8i2.821.

[9] R. Andriansyah *et al.*, "Augmented reality approach to introduce batik Garutan using features accelerated segment test," in *2021 7th International Conference on Wireless and Telematics (ICWT)*, 2021, pp. 1–5, doi: 10.1109/ICWT52862.2021.9678417.

[10] A. E. Minarno, F. D. S. Sumadi, H. Wibowo, and Y. Munarko, "Classification of batik patterns using K-Nearest neighbor and support vector machine," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1260–1267, 2020, doi: 10.11591/eei.v9i3.1971.

[11] M. A. Rasyidi and T. Bariyah, "Batik pattern recognition using convolutional neural network," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1430–1437, 2020, doi: 10.11591/eei.v9i4.2385.

[12] D. A. Mardani, Pranowo, and A. J. Santoso, "Deep learning for recognition of Javanese batik patterns," in *The 5th international conference on industrial, mechanical, electrical, and chemical engineering 2019 (ICIMECE 2019)*, AIP Publishing, 2020, pp. 1-7, doi: 10.1063/5.0000686.

[13] I. Nurhaida, V. Ayumi, D. Fitrianah, R. A. M. Zen, H. Noprisson, and H. Wei, "Implementation of deep neural networks (DNN) with batch normalization for batik pattern recognition," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 2, pp. 2045-2053, 2020, doi: 10.11591/ijece.v10i2.pp2045-2053.

[14] Y. Azhar, M. C. Mustaqim, and A. E. Minarno, "Ensemble convolutional neural network for robust batik classification," *IOP*

Conference Series: Materials Science and Engineering, vol. 1077, no. 1, 2021, doi: 10.1088/1757-899x/1077/1/012053.

[15] C. U. Khasanah, E. Utami, and S. Raharjo, "Implementation of data augmentation using convolutional neural network for batik classification," in *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, IEEE, Oct. 2020, pp. 1–5, doi: 10.1109/CITSM50537.2020.9268890.

[16] Z. Widyantoko, T. P. Widowati, I. Isnaini, and P. Trapsiladi, "Expert role in image classification using CNN for hard to identify object: distinguishing batik and its imitation," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 1, pp. 93-100, 2021, doi: 10.11591/ijai.v10.i1.pp93-100.

[17] N. W. P. Septiani *et al.*, "Convolutional neural network (CNN) algorithm for geometrical batik sade' motifs," in *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, IEEE, Feb. 2023, pp. 597–602, doi: 10.1109/ICCoSITE57641.2023.10127829.

[18] D. G. T. Meranggi, N. Yudistira, and Y. A. Sari, "Batik classification using convolutional neural network with data improvements," *JOIV : International Journal on Informatics Visualization*, vol. 6, no. 1, pp. 6-11, Mar. 2022, doi: 10.30630/joiv.6.1.716.

[19] R. R. Isnanto, A. Hidayatno and A. A. Zahra, "Fractal batik motifs generation using variations of parameters in julia set function," *2020 8th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia,* 2020, pp. 1-6, doi: 10.1109/ICoICT49345.2020.9166282.

[20] N. D. Girsang, "Literature study of convolutional neural network algorithm for batik classification," *Brilliance: Research of Artificial Intelligence*, vol. 1, no. 1, pp. 1–7, Sep. 2021, doi: 10.47709/brilliance.v1i1.1069.

[21] W. Herulambang, M. N. Hamidah and F. Setyatama, "Comparison of SVM And BPNN methods in the classification of batik patterns based on color histograms and invariant moments," *2020 International Conference on Smart Technology and Applications (ICoSTA), Surabaya, Indonesia,* 2020, pp. 1-4, doi: 10.1109/ICoSTA48221.2020.1570615583.

[22] J. Tristanto, J. Hendryli, and D. E. Herwindiati, "Classification of batik motifs using convolutional neural networks," *SSRN Electronic Journal*, pp. 1-5, 2018, doi: 10.2139/ssrn.3258935.

[23] H. Prasetyo and B. A. P. Akardihas, "Batik image retrieval using convolutional neural network," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 6, pp. 3010-3018, 2019, doi: 10.12928/telkomnika.v17i6.12701.

[24] A. A. Kasim, R. Wardoyo, A. Harjoko, "Batik classification with artificial neural network based on texture-shape feature of main ornament," *International Journal of Intelligent Systems and Applications (IJISA)*, vol. 9, no. 6, pp. 55–65, 2021, doi: 10.5815/ijisa.2017.06.06.

[25] A. H. Rangkuti, M. M. Ramadhan and A. H. Aslamia, "Reliable batik image classification: mulwin-LBP algorithm and deep neural network, " *2021 3rd International Symposium on Material and Electrical Engineering Conference (ISMEE)*, Bandung, Indonesia, 2021, pp. 96-101, doi: 10.1109/ISMEE54273.2021.9774230.

BIOGRAPHIES OF AUTHORS

Rosalina is an esteemed lecturer in the Informatics Study Program at President University, exemplifies dedication to her field. Holding a Master's degree in Informatics from President University underscores her commitment to academic excellence. With her strong educational foundation and passion for informatics, Rosalina significantly influences students' academic journeys. Her adeptness in the subject matter, coupled with her skill in conveying complex concepts, cultivates a dynamic and enriching learning atmosphere. She can be contacted at email: rosalina@president.ac.id.

Genta Sahuri a dedicated lecturer at President University's Information Systems Study Program, brings a wealth of knowledge and expertise to his role. Holding a Master's degree in Informatics from the same institution underscores his commitment to academic excellence. With a strong educational background and a passion for his field, he plays a crucial role in shaping the academic journey of his students. His adeptness in conveying complex concepts fosters a dynamic and enriching learning environment. Genta's dedication to his field and his ability to inspire students make him a valuable asset to President University. He can be contacted at email: genta.sahuri@president.ac.id.

Hana Desvira is a student enrolled in the Informatics Study Program at President University. Driven by a fervor for cutting-edge technology, her research focus centers on the dynamic realm of machine learning. She is devoted to delving into artificial intelligence disciplines and harnessing its capabilities to tackle real-world problems. Her enthusiasm and dedication make her a valuable asset to the academic community. She can be contacted at email: hana.desriva@student.president.ac.id.