2649-9689-1-PB.pdf

Institut Teknologi Dirgantara Adisutjipto

Document Details

Submission ID

trn:oid:::3618:79332748

Submission Date

Jan 16, 2025, 2:42 PM GMT+7

Download Date

Jan 16, 2025, 3:09 PM GMT+7

File Name

2649-9689-1-PB.pdf

File Size

276.8 KB

8 Pages

4,076 Words

23,063 Characters

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 10 words)

Exclusions

- 3 Excluded Matches

Match Groups

18 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5%	Internet sources
2%	Publications
7%	Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

Hidden Text
9 suspect characters on 1 page
Text is altered to blend into the white background of the document.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

18 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5%	Internet sources
2%	Publications
7%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	International Business School on 2024-11-19	2%
2	Internet	ejournals.itda.ac.id	2%
3	Submitted works	University of Greenwich on 2024-07-11	<1%
4	Internet	www.jazindia.com	<1%
5	Submitted works	Mepco Schlenk Engineering college on 2024-04-15	<1%
6	Internet	wjaets.com	<1%
7	Internet	www.nature.com	<1%
8	Submitted works	Federal University of Technology on 2024-01-26	<1%
9	Submitted works	The University of Memphis on 2024-11-20	<1%
10	Internet	repository.ittelkom-pwt.ac.id	<1%

11	Submitted works	Deakin University on 2021-05-20	<1%
12	Submitted works	SSN COLLEGE OF ENGINEERING, Kalavakkam on 2023-10-15	<1%
13	Internet	journalofbigdata.springeropen.com	<1%
14	Submitted works	South Bank University on 2025-01-03	<1%
15	Submitted works	University of Hertfordshire on 2024-04-30	<1%
16	Internet	repository.iiitd.edu.in	<1%

COMPILER
Vol. 13, No. 2, November 2024, pp. 123-130
ISSN: 2252-2403 (Print), ISSN: 2549-2403 (Online)
Aaccredited Fourth Grade by Kemenristekdikti, Decree No:200/M/KPT/2020
DOI: 10.28989/compiler.v13i2.2649

Improving the Accuracy of Batik Classification using Deep Convolutional Auto Encoder

Muhammad Faqih Dzulqarnain[1,2,*], Abdul Fadlil[3], Imam Riadi[4]
[1]Departement of Informatics, Universitas Ahmad Dahlan, Indonesia
[2]Department of Information Technology, Politeknik Aisyiyah Pontianak, Indonesia
[3]Department of Electrical Engineering, Universitas Ahmad Dahlan, Indonesia
[4]Department of Information System, Universitas Ahmad Dahlan, Indonesia

Article Info

Article history:

Received December 3, 2024
Accepted December 9, 2024
Published December 20, 2024

Keywords:

Autoencoder
DCAE
Accuracy
Batik Classification

ABSTRACT

Batik patterns are a reference for the classification of batik origin and the culture depicted in it. Several studies have raised the classification of batik patterns but have gaps to be developed in terms of classification accuracy. This research investigates the development of model deep convolutional autoencoders to enhance the classification of digital batik images. The dataset used was sourced from Kaggle. The autoencoder was employed to enrich the image data prior to convolutional processing. By forcing the autoencoder to learn a lower-dimensional latent representation that captures the most salient features of the batik patterns. The performance of this enhanced model was compared against a standard convolutional neural network (CNN) without the autoencoder. Experimental results demonstrate that the incorporation of the autoencoder significantly improved the classification accuracy, achieves 99% accuracy on test data with a total of 30% of 680 images data, 99.5% accuracy on training data with a total of 70% of 680 images data, loss value of 3.4% from data testing, and loss value of 3,8% from data training. This study highlights the potential of deep convolutional autoencoders as a powerful tool for augmenting image data and improving the performance of deep learning models in the context of batik image classification.

Corresponding Author:

Muhammad Faqih Dzulqarnain,
Department of Informatics,
Universitas Ahmad Dahlan,
Jl. Ringroad Selatan, Kabupaten Bantul, Daerah Istimewa Yogyakarta, Indonesia
Email: [1]2437083007@webmail.uad.ac.id[*]

1. INTRODUCTION

Indonesia boasts a rich tapestry of cultural and artistic heritage, with batik being a prime example [1]. As an intangible cultural heritage of Indonesia, batik holds significant value in representing the identity and philosophy of a region [2]. West Kalimantan, with its diverse ethnicities and cultures, produces a variety of unique and meaningful batik motifs [3]. These motifs, such as Corak Insang, Ikat Celup, Megamendung, and indigenous Dayak motifs, reflect the interaction between humans and the environment [4]. The spiritual values cherished by society also give batik a distinctive character in Indonesia [5].

The digital world presents opportunities to develop new technologies for studying and preserving cultural heritage, including batik [6]. Studies on batik pattern recognition have been conducted and have contributed to the development of pattern recognition technology [7]. The utilization of digital image technology, particularly the development of Convolutional Neural Network (CNN) models, has the potential to advance these efforts [8]. CNNs operate by mimicking the visual cortex of the human brain, allowing them to "see" and "understand" patterns in digital images [9]. Inspired by the human neural system, CNNs have demonstrated their capabilities in various pattern recognition tasks, including image classification [10]. Several studies have provided good results on the accuracy of batik image classification, but the accuracy value can still be improved to avoid data classification errors when tested with new data by developing a new CNN model.

The implementation of CNNs in the context of batik pattern recognition can be directed towards automatic identification and classification of batik motifs [11]. Developing CNN model implementations enables the construction of a comprehensive, structured batik motif database with high accuracy [12]. This model development can be achieved by applying Autoencoders to the convolutional process of CNN [13]. Autoencoders serve as a reference for improving model learning outcomes to enhance pattern learning [14].

The objective of this research is to develop a CNN model with Autoencoder, resulting in a Deep Convolutional Autoencoder (DCAE), to improve the accuracy [15] of batik pattern classification. The Autoencoder will be applied to produce sharper image processing results [16][17] to avoid training and testing errors during the convolution process [18]. While Autoencoders have been applied in various studies, their application to batik pattern learning is novel.

2. RESEARCH METHOD

The methodology employed in this research is the Deep Convolutional Autoencoder (DCAE), as illustrated in Figure 1. The DCAE is a type of artificial neural network that combines two architectures for image data processing: convolutional neural networks (CNNs) and autoencoders [19]. DCAE is designed to automatically learn efficient feature representations from input data, typically images, through an encoding and decoding process [20]. It is utilized for unsupervised learning, dimensionality reduction, and image data reconstruction [21]. The DCAE aims to learn an efficient latent representation (code) of the input data, which can subsequently be used to reconstruct the original data or for other tasks such as denoising [22].

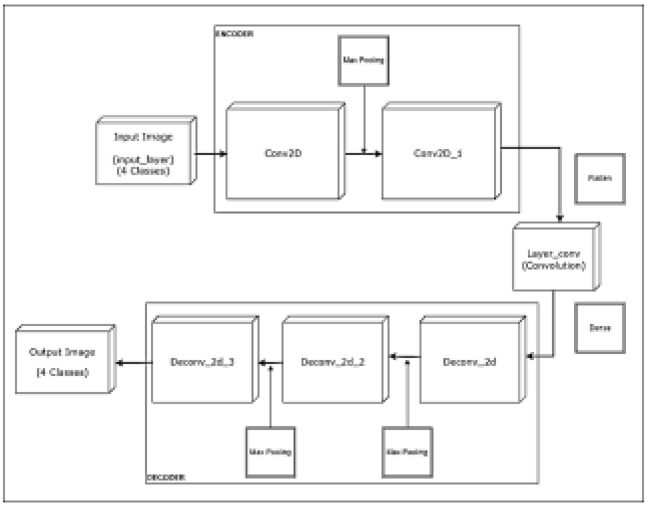


Figure 1. Diagram of DCAE Model Design

Figure 1 illustrates the process of a Deep Convolutional Autoencoder (DCAE). The DCAE is a neural network architecture designed to learn efficient representations of input data, typically images, through an encoding and decoding process.

1. Input Image: The original image that serves as the input to the DCAE.
2. Conv 1, Conv 2, Conv 3: These are convolutional layers within the encoder that extract features from the input image. As the network goes deeper (Conv 2, Conv 3), the extracted features become more abstract and hierarchical, capturing higher-level semantic information [23].
3. Latent Representation: This is a compressed representation of the input image, often referred to as a "code." It is the bottleneck of the autoencoder and contains the most essential information extracted from the input. The latent representation can be used for various tasks, such as dimensionality reduction, feature extraction, and denoising [24][25].
4. Deconv 3, Deconv 2, Deconv 1: These are transposed convolutional layers within the decoder that reconstruct the image from the latent representation. They reverse the process of the encoder, gradually building up a higher-resolution image from the learned features [23].
5. Output Image: The reconstructed image generated by the DCAE. Ideally, it should be as similar as possible to the original input image.

The research objects were obtained from the Kaggle dataset [4], consisting of 680 batik image data. These image data were categorized into several classes as presented in Table 1.

Table 1. Digital Image Objects for Dataset

No	Classes	Amount
1	Batik Corak Insang	170 Images
2	Batik Dayak	170 Images
3	Batik Megamendung	170 Images
4	Batik Ikat Celup	170 Images
Total		680 Images

The image dataset was divided into training and testing subsets, with 70% of the data allocated for data training and 30% for data testing.

Table 2. Dataset Division into Training Data and Testing Data

No	Classes	Amount
1	Data Training (70%)	476 Images
2	Data Testing (30%)	204 Images

All images were preprocessed by resizing them to a standardized dimension of 224x224 pixels. This resolution was chosen to balance computational efficiency and the preservation of essential pattern details.

3. RESULTS AND ANALYSIS

The research commenced with a dataset of batik images procured from Kaggle [4]. To augment the training data and improve the model's generalization capabilities, horizontal and vertical flips were applied to the images. This data augmentation technique introduced additional variations in the training set, enabling the model to learn more robust features. Subsequently, an autoencoder model was constructed as detailed in Table 3. The autoencoder's architecture was designed to efficiently extract latent representations from the input images. The encoder component of the autoencoder consisted of convolutional layers that progressively down sampled the input images, while the decoder component sampled the latent representation to reconstruct the original image. This architecture allowed the model to learn a compressed representation of the batik patterns, capturing the underlying semantic information.

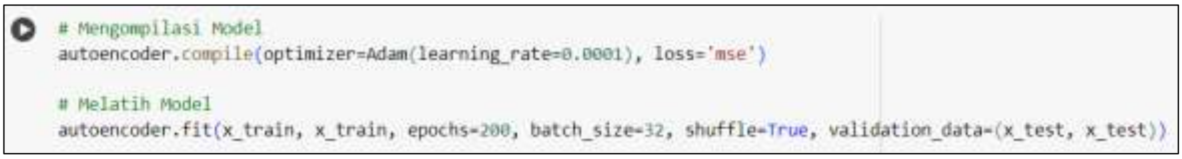
Table 3. Summary of Autoencoder Model Design

No	Layer (type)	Output Shape	Param #
1	input_layer (InputLayer)	(None, 100, 100, 3)	0
2	conv2d (Conv2D)	(None, 100, 100, 12)	896
3	max_pooling2d (MaxPooling2D)	(None, 50, 50, 32)	0
4	conv2d_1 (Conv2D)	(None, 50, 50, 64)	18.496
5	max_pooling2d_1 (MaxPooling2D)	(None, 25, 25, 64)	0
6	conv2d_2 (Conv2D)	(None, 25, 25, 128)	73.856
7	max_pooling2d_2 (MaxPooling2D)	(None, 13, 13, 128)	0
8	conv2d_3 (Conv2D)	(None, 13, 13, 128)	147.584
9	up_sampling2d (UpSampling2D)	(None, 26, 26, 128)	0
10	conv2d_4 (Conv2D)	(None, 26, 26, 64)	73.792
11	up_sampling2d_1 (UpSampling2D)	(None, 52, 52, 64)	0
12	conv2d_5 (Conv2D)	(None, 50, 50, 32)	18.464
13	up_sampling2d_2 (UpSampling2D)	(None, 100, 100, 32)	0
14	conv2d_6 (Conv2D)	(None, 100, 100, 3)	867

Table 3 presents a detailed overview of the proposed autoencoder architecture. The encoder component, depicted from number 1 to 4 in table 3, comprises a series of convolutional layers with decreasing spatial dimensions and increasing numbers of filters. This progressive down sampling captures hierarchical features from the input image, culminating in a latent representation. Conversely, the decoder, highlighted from number 5 to 14 in table 3, mirrors the encoder's structure but in reverse, gradually reconstructing the image from the latent code. The decoder employs transposed convolutions to up sample the feature maps, enabling the generation of an output image with the same dimensions as the input.

3.1. Training the Model

The autoencoder model was trained on the augmented dataset using a suitable loss function, such as mean squared error (MSE), to minimize the reconstruction error between the input and output images. During training, the model learned to encode the input images into a lower-dimensional latent space and then decode this latent representation to reconstruct the original image. The latent representations captured the most salient features of the batik patterns, providing a compact and informative representation of the data. Once the autoencoder was trained, the convolutional features extracted by the encoder were used as input to a subsequent convolutional neural network (CNN) for the classification task and train the classification (Figure 2).

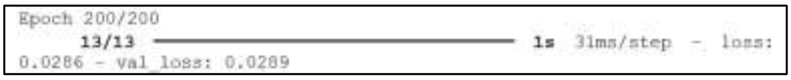


```
# Mengompilasi Model
autoencoder.compile(optimizer=Adam(learning_rate=0.0001), loss='mse')

# Melatih Model
autoencoder.fit(x_train, x_train, epochs=200, batch_size=32, shuffle=True, validation_data=(x_test, x_test))
```

Figure 2. Code for Training DCAE Design Model

The convolutional features obtained from the autoencoder were fed into a CNN to perform the final classification task. The CNN was designed with multiple convolutional layers, followed by pooling layers to progressively reduce the spatial dimensions of the feature maps. The output of the final convolutional layer was flattened and fed into a fully connected layer, which produced the final classification probabilities. The CNN was trained using a categorical cross-entropy loss function to optimize the model's parameters. The training process involved iteratively updating the model's weights to minimize the classification error on the training set. To evaluate the performance of the model, it was tested on a held-out test set, and the classification accuracy was computed.



```
Epoch 200/200
     13/13 ━━━━━━━━━━━━━━━━━━━━ 1s 31ms/step - loss:
0.0286 - val_loss: 0.0289
```

Figure 3. Results of model training

Figure 3 describes the DCAE was trained over 200 epochs, resulting in a converged loss and validation loss of 2.8%. The low loss values suggest that the model has successfully learned to reconstruct the input images with high fidelity. This indicates that the latent space learned by the autoencoder captures the underlying structure and variations within the batik dataset.

3.2. Combination of DCAE and CNN

After the training of the DCAE, the latent representations obtained from the encoder were utilized as input to a convolutional neural network (CNN). This model is a development from standard CNN to learn high-level features on the DCAE latent space, which are more discriminative for the classification task. CNN was trained for 200 epochs using a suitable loss function, such as categorical cross-entropy, to optimize the classification performance. A summary of the classifier model's architecture and performance metrics is provided in Table 3, showcasing the model's ability to accurately classify batik patterns.

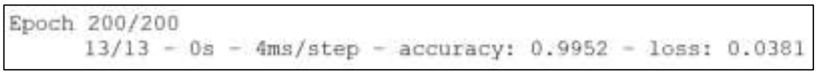
Table 4. Summary of CNN Model

No	Layer (type)	Output Shape	Param #
1	input_layer_3 (InputLayer)	(None, 13, 13, 128)	0
2	flatten_2 (Flatten)	(None, 21.632)	0
3	dense_6 (Dense)	(None, 128)	2.769.024
4	dense_7 (Dense)	(None, 4)	516

Table 3 presents a concise overview of the classifier model, including its key components and performance evaluation. As shown in the table, the model achieved a high classification accuracy on the test set.

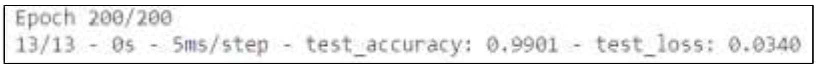
3.3. Training and Testing Process

The combination of DCAE and CNN for image classification shows excellent performance as shown in figure 4, attaining a validation accuracy of 99,5% for testing with a corresponding loss of 3,8% and for training make 99% accuracy and loss 3,4% in figure 5.



```
Epoch 200/200
     13/13 - 0s - 4ms/step - accuracy: 0.9952 - loss: 0.0381
```

Figure 4. Combined model classification results for training



```
Epoch 200/200
13/13 - 0s - 5ms/step - test_accuracy: 0.9901 - test_loss: 0.0340
```

Figure 5. Combined model classification results for testing

These results indicate that the model has effectively learned to discriminate between different classes of batik patterns. This result can also be seen from the plot diagram of figure 6 for training and figure 7 for testing.

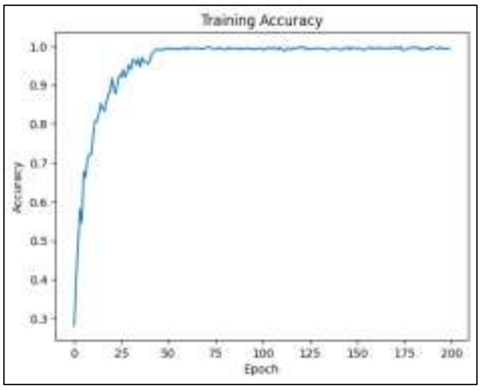


Figure 6. Training Data Results Diagram

Figure 6 presents the learning curves obtained by training both the DCAE and CNN models. The consistently high accuracy achieved by both models across multiple epochs indicates their robust ability to learn intricate patterns within the training data. This suggests that these models have the potential to generalize well to unseen data and perform accurate pattern recognition tasks.

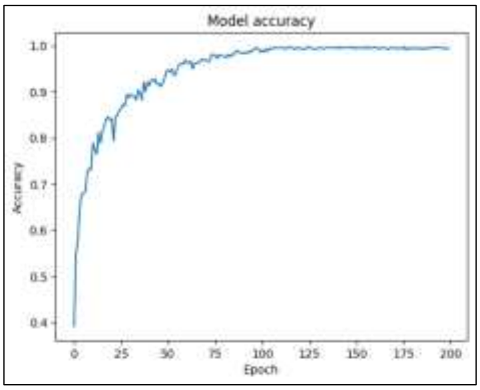


Figure 7. Testing Data Results Diagram

The test curve in figure 7 shows strong and stable convergence, comparable to the monotonically decreasing loss training in figure 8. This indicates that the model has learned a generalizable representation of the data.

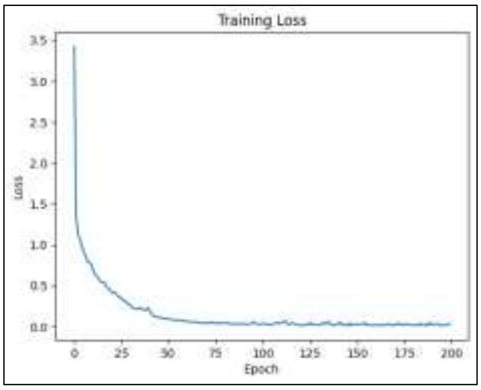


Figure 8. Loss Training Results Diagram

The training loss plot in Figure 8 demonstrates the stability of the combined DCAE and CNN model during training. The steady decline in the loss value suggests that the model is converging to a good solution and is not overfitting to the training data. This indicates the robustness of the combined model and its potential for generalization to unseen data.

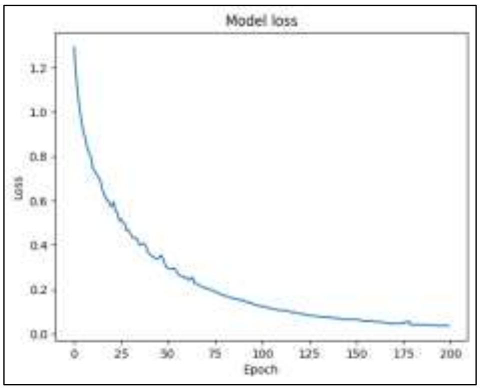


Figure 9. Loss Testing Results Diagram

Figure 9 describes the model achieving a final variance loss of 3.8% after more than 200 epochs. The loss function showed a steady decrease during training, indicating that the model did not suffer from data training failure and was able to effectively learn the underlying patterns in the data.

3.4. Evaluate the Model with Confusion Matrix

Evaluation of the proposed model uses a confusion matrix as shown in Figure 10 and Figure 11. The figure shows the model's ability to learn effectively and generalize well to unseen data, with minimal signs of overfitting.

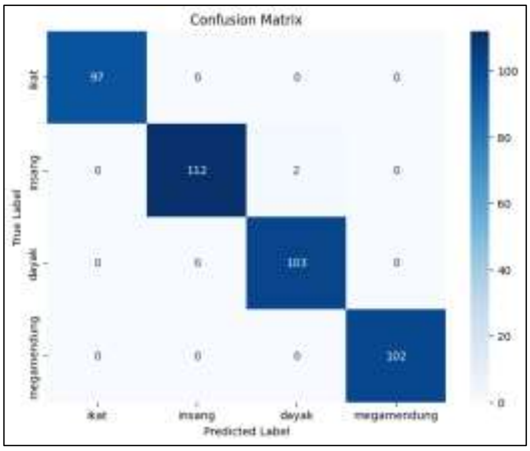


Figure 10. Confusion Matrix Model Training Results

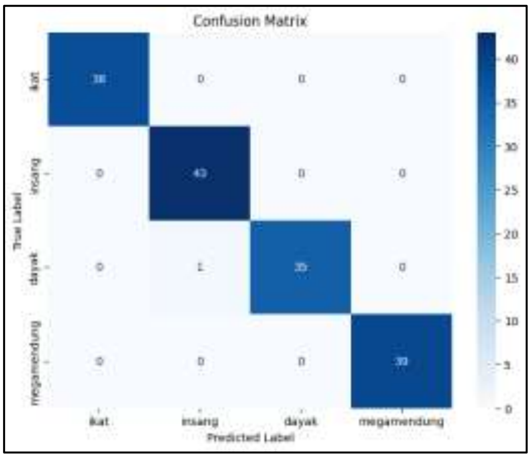


Figure 11. Confusion Matrix Model Testing Results

The confusion matrix in figure 10 and figure 11 provides a detailed breakdown of the classifier's performance from data training (Figure 10) and data testing (Figure 11). The model achieved an overall accuracy of 99,5%, f1-score 99,5% and recall with 99,5%, indicating high performance in classifying batik patterns. In particular, the classifier performs very well in distinguishing between each class in the batik images, with only a few misclassifications. Nevertheless, the overall results demonstrate the effectiveness of the proposed DCAE model in classifying batik patterns.

3.4. Comparison with Previous Research

This section presents a comparative analysis of the accuracy achieved by the proposed DCAE model and pre-existing CNN models for batik pattern recognition with same dataset [4][8][12]. By directly comparing the classification performance of these models, this research aims to assess the effectiveness of the DCAE architecture in capturing features of batik patterns and outperforming traditional CNN approaches. The DCAE model, designed to learn deep latent representations of batik patterns, is compared with conventional CNN as shown in Table 4.

Table 5. Comparison of Previous Research Results

No	Research	Accuracy
1	Implementasi CNN with Transfer Learning [4]	70 %
2	CNN with Transfer Learning + VGG [8]	91,23%
3	CNN back & forward propagation [12]	91,24%
4	DCAE Model (This research)	99%

Comparison from table 4 to investigate the impact of different CNN architectures on batik pattern recognition. This study presents a novel CNN architecture for batik pattern recognition, achieving a state-of-the-art accuracy of 99.5%. This model significantly outperforms existing approaches, such as the baseline CNN (70% accuracy), CNN with VGG (91.23% accuracy), and CNN with back and forward propagation (91.24%).

The superior performance of this new model can be attributed to its ability to learn a deep latent representation of batik patterns while preserving the dimensionality of the input data. By using an autoencoder architecture, the DCAE can effectively capture the spatial relationships between pixels, enabling it to learn more informative features compared to traditional CNNs. This, coupled with the model's ability to reconstruct the input images, contributes to its improved classification performance.

4. CONCLUSION

This study introduces a novel CNN architecture, DCAE, for batik pattern recognition. The proposed model achieves value accuracy of 99.5% from data training with 70% of 680 images data and 99% from data testing with 30% of 680 images data on the batik pattern recognition task. The new model significantly outperforms previous studies that have used various CNN architectures for this task. The superior performance of DCAE can be attributed to its unique architecture. Utilizing an autoencoder, DCAE learns a deep latent representation of the input data. This latent representation effectively captures the intricate details and variations in the batik patterns, allowing the model to make more accurate classifications. In addition, the autoencoder component enables DCAE to denoise the input images, reducing the impact of noise and artifacts that can hinder classification performance. Unlike traditional CNNs, which primarily focus on extracting features from input images, DCAE also learns to reconstruct the input data. This reconstruction process helps the model learn more robust and generalizable features, thereby improving its ability to classify unseen data.

Future research could focus on enhancing the DCAE model by delving deeper into data validation. Specifically, efforts should be directed towards achieving lower loss values and expanding the dataset to include more varied and complex patterns.

REFERENCES

[1] A. Kusumastuti, Atika, T. A. Achmadi, K. Phusavat, and A. N. Hidayanto, “Assessment of producer’s perspective on the production of environmentally friendly fashion products: a case study in Indonesian natural dyes batik craftsmen,” *Environmental Science and Pollution Research*, vol. 30, no. 60, 2023, doi: 10.1007/s11356-022-23330-z.

[2] Aquarini, Ishomuddin, V. Salviana DS, and F. M, “The Social Meaning of the Enggang Bird in Batik of Dayak Community of Central Kalimantan, Indonesia,” *International Journal of Humanities, Social Sciences and Education*, vol. 9, no. 5, 2022, doi: 10.20431/2349-0381.0905007.

[3] R. Wiryadinata, M. R. Adli, R. Fahrizal, and R. Alfanz, “Klasifikasi 12 Motif Batik Banten Menggunakan Support Vector Machine,” *Jurnal EECCIS*, vol. 13, no. 1, 2019.

[4] I. Maulana, H. Sastypratiwi, H. Muhardi, N. Safriadi, and H. Sujaini, “Implementasi Convolutional Neural Network (CNN) untuk Klasifikasi Motif Batik pada Aplikasi Computer Vision Berbasis Android,” *JEPIN - Jurnal Edukasi dan Penelitian Informatika*, vol. 9, no. 3, pp. 384–393, 2023.

[5] D. G. T. Meranggi, N. Yudistira, and Y. A. Sari, “Batik Classification Using Convolutional Neural Network with Data Improvements,” *International Journal on Informatics Visualization*, vol. 6, no. 1, 2022, doi: 10.30630/joiv.6.1.716.

[6] A. Tejawati, J. A. Widians, R. Sulle, Muhammad Bambang Firdaus, A. Prafanto, and F. Alameka, “Pemodelan Konsep Augmented Reality Motif Batik Dayak Kalimantan Timur,” *METIK JURNAL*, vol. 6, no. 1, 2022, doi: 10.47002/metik.v6i1.333.

[7] 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, 2022, doi: 10.26555/ijain.v8i2.821.

[8] R. F. Alya, M. Wibowo, and P. Paradise, “Classification of Batik Motif Using Transfer Learning on Convolutional Neural Network (CNN),” *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 1, 2023, doi: 10.52436/1.jutif.2023.4.1.564.

[9] K. Azmi, S. Defit, and Sumijan, “Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat,” *Jurnal Unitek*, vol. 16, no. 1, pp. 28–40, 2023.

[10] A. Prayoga, Maimunah, P. Sukmasetya, Muhammad Resa Arif Yudianto, and Rofi Abul Hasani, “Arsitektur Convolutional Neural Network untuk Model Klasifikasi Citra Batik Yogyakarta,” *Journal of Applied Computer Science and Technology*, vol. 4, no. 2, 2023, doi: 10.52158/jacost.v4i2.486.

[11] N. W. Parwati Septiani *et al.*, “Convolutional Neural Network (CNN) Algorithm for Geometrical Batik Sade’ Motifs,” in *ICCoSITE 2023 - International Conference on Computer Science, Information Technology and Engineering: Digital Transformation Strategy in Facing the VUCA and TUNA Era*, 2023. doi: 10.1109/ICCoSITE57641.2023.10127829.

[12] M. M. A. Wona *et al.*, “Klasifikasi Batik Indonesia Menggunakan Convolutional Neural Network (CNN),” *JURTI*, vol. 7, no. 2, pp. 172–179, 2023, [Online]. Available: https://www.kaggle.com/datasets/dionisiusdh/indonesianbatik-motifs.

[13] P. Bortnowski, H. Gondek, R. Król, D. Marasova, and M. Ozdoba, “Detection of Blockages of the Belt Conveyor Transfer Point Using an RGB Camera and CNN Autoencoder,” *Energies (Basel)*, vol. 16, no. 4, 2023, doi: 10.3390/en16041666.

[14] W. Kurniawan, Y. Kristian, and J. Santoso, “Pemanfaatan Deep Convolutional Auto-encoder untuk Mitigasi Serangan Adversarial Attack pada Citra Digital,” *J-INTECH(Journal ofInformation and Technology)*, vol. 11, no. 1, pp. 50–59, 2023.

[15] H. M. Tornyeviadzi and R. Seidu, “Leakage detection in water distribution networks via 1D CNN deep autoencoder for multivariate SCADA data,” *Eng Appl Artif Intell*, vol. 122, 2023, doi: 10.1016/j.engappai.2023.106062.

[16] K. N. Sunil Kumar, G. B. Arjun Kumar, R. Gatti, S. Santosh Kumar, D. A. Bhyratae, and S. Palle, “Design and implementation of auto encoder based bio medical signal transmission to optimize power using convolution neural network,” *Neuroscience Informatics*, vol. 3, no. 1, p. 100121, Mar. 2023, doi: 10.1016/j.neuri.2023.100121.

[17] F. Deng, W. Luo, B. Wei, Y. Zuo, H. Zeng, and Y. He, “A novel insulator defect detection scheme based on Deep Convolutional Auto-Encoder for small negative samples,” *High Voltage*, vol. 7, no. 5, 2022, doi: 10.1049/hve2.12210.

[18] R. Zhao, Z. Yang, X. Meng, and F. Shao, “A Novel Fully Convolutional Auto-Encoder Based on Dual Clustering and Latent Feature Adversarial Consistency for Hyperspectral Anomaly Detection,” *Remote Sens (Basel)*, vol. 16, no. 4, 2024, doi: 10.3390/rs16040717.

[19] Z. Qiang, L. He, F. Dai, Q. Zhang, and J. Li, “Image inpainting based on improved deep convolutional auto-encoder network,” *Chinese Journal of Electronics*, vol. 29, no. 6, 2020, doi: 10.1049/cje.2020.09.008.

[20] V. Turchenko, E. Chalmers, and A. Luczak, “A deep convolutional auto-encoder with pooling - unpooling layers in caffe,” *International Journal of Computing*, vol. 18, no. 1, 2019, doi: 10.47839/ijc.18.1.1270.

[21] H. E. Atlason, A. Love, S. Sigurdsson, V. Gudnason, and L. M. Ellingsen, “SegAE: Unsupervised white matter lesion segmentation from brain MRIs using a CNN autoencoder,” *Neuroimage Clin*, vol. 24, 2019, doi: 10.1016/j.nicl.2019.102085.

[22] O. B. Ozdemir and A. Koz, “3D-CNN and Autoencoder-Based Gas Detection in Hyperspectral Images,” *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 16, 2023, doi: 10.1109/JSTARS.2023.3235781.

[23] X. Song *et al.*, “A Semantic Segmentation Method for Road Environment Images Based on Hybrid Convolutional Auto-Encoder,” *Traitement du Signal*, vol. 39, no. 4, 2022, doi: 10.18280/ts.390416.

[24] Y. Farooq and S. Savas, “Noise Removal from the Image Using Convolutional Neural Networks-Based Denoising Auto Encoder,” *Journal of Emerging Computer Technologies*, vol. 3, no. 1, 2024, doi: 10.57020/ject.1390428.

[25] J. Li, J. Wang, and Z. Lin, “SGCAST: symmetric graph convolutional auto-encoder for scalable and accurate study of spatial transcriptomics,” *Brief Bioinform*, vol. 25, no. 1, 2024, doi: 10.1093/bib/bbad490.

[26] D. Yang, H. R. Karimi, and K. Sun, “Residual wide-kernel deep convolutional auto-encoder for intelligent rotating machinery fault diagnosis with limited samples,” *Neural Networks*, vol. 141, pp. 133–144, 2021, doi: https://doi.org/10.1016/j.neunet.2021.04.003.

[27] A. W. Murdiyanto and M. Habibi, “Analysis of Deep Learning Approach Based on Convolution Neural Network (CNN) for Classification of Web Page Title and Description Text,” *Compiler*, vol. 11, no. 2, pp. 51–58, 2022.