Identification of Batik in Central Java using Transfer Learning Method

Stephanie Pamela Adithama*[1], B. Yudi Dwiandiyanta[2], Sevia Berliana Wiadji[3]
Program Studi Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta, Sleman 55281, Indonesia
Email: [1]stephanie.pamela@uajy.ac.id, [2]yudi.dwiandiyanta@uajy.ac.id, [3]180709593@students.uajy.ac.id

Abstrak. Identifikasi Batik di Jawa Tengah dengan Metode Transfer Learning. *Batik diakui sebagai warisan kemanusiaan untuk budaya lisan dan non bendawi oleh UNESCO karena ikatan simbolis dan filosofisnya dengan kehidupan masyarakat Indonesia. Namun, generasi muda secara bertahap kehilangan warisan mereka sebagai akibat dari perubahan teknologi dan sosiologis yang mempengaruhi batik Indonesia. Konsekuensinya, menyebabkan lunturnya pengetahuan tentang batik. Maka dari itu, dibangunlah model untuk membantu identifikasi motif batik menggunakan convolutional neural network memanfaatkan teknik transfer learning dari deep learning. Penelitian ini dilakukan dengan memanfaatkan dataset yang terdiri dari 1000 gambar, lima kelas motif batik (Banji, Kawung, Lereng, Parang, dan Slobog), dan menggunakan pre-trained model VGG16 dan VGG19 pada Keras. Model terbaik \yaitu menggunakan arsitektur VGG16 dan jumlah epochs 50 dengan hasil testing accuracy sebesar 0,9200.*
Keywords: *batik, transfer learning, Convolutional Neural Network, deep learning, VGG16*

Abstract. Identification of Batik in Central Java using Transfer Learning Method. *Batik was recognized as a human heritage for oral and nonmaterial culture by UNESCO due to its symbolic and philosophical ties to the lives of Indonesians. However, the younger generation is gradually losing its legacy because of technological and sociological changes that have influenced Indonesian batik. Consequently, batik knowledge is disappearing. A convolutional neural network and transfer learning techniques were utilized in deep learning to construct a model recognising batik motifs. The study utilized a dataset of one thousand images, five classes of batik designs (Banji, Kawung, Slope, Parang, and Slobog), and pre-trained architectural models VGG16 and VGG19 on Keras. The best model utilizes the VGG16 architecture, and the number of epochs is 50, with the result of testing accuracy of 0.9200.*
Keywords: *batik, transfer learning, Convolutional Neural Network, deep learning, VGG16*

1. Introduction

Batik is a highly appreciated traditional art style used for centuries in numerous Asian and African nations [1]. Indonesia is an Asian country with a batik culture. According to KBBI, batik is a type of textile with a picture created explicitly by writing on or adding wax to the fabric before being processed in a particular method [2]. Indonesian batik was designated as a human heritage for oral and nonmaterial culture by UNESCO in 2009 because it has its own symbolic and philosophical underpinnings relating to the lives of Indonesians [3][4]. As a traditional product, batik embodies philosophic ideas in its design and functionality [5]. However, like other traditional handicrafts, Indonesian batik has been affected by technological and societal developments and is suffering a development problem, with the younger generation gradually losing their inheritance [6][7]. Batik has started to lose popularity in Indonesia itself, particularly among teenagers. Younger Indonesians favor utilizing international fashion trends [8]. Because there are so many distinct batik themes, many are unaware of the name of a particular motif. Even though there is data and information on batik themes, it is still difficult to find pertinent information with various motifs. The issue is to raise millennials' cultural, historical, and geographical awareness of batik [9][10]. Therefore, technology must be utilized to resolve the issue.

Creating a machine that can recognize batik motifs and deliver information about these motifs is one of the technologies that can assist in solving existing challenges. This will facilitate the younger generation's access to information through technological means. The concepts of deep learning can be used to construct this system, which will result in high levels of accuracy. Deep learning employs multiple layers to develop its computational processes. An artificial neural network can be used as the layer in deep learning to address problems posed by large datasets [11]. The deep learning method with the most notable results for image recognition and classification is the convolutional neural network (CNN) [12]. Image deformation can be handled using CNN, which also helps to minimize the number of unneeded parameters [13]. Object classification, scene identification, and object detection are just a few of the powerful visual representation capabilities that CNNs possess that are used for recognition at all levels of categories.

This study describes the process of recognizing batik motifs in Central Java, beginning with the existing problems in the field. In contrast to previous studies, the current study intends to increase the accuracy of models utilized in earlier research. It is intended that by enhancing accuracy, the identification process will operate smoothly. In the end, the objective of using technology to provide accurate information to future generations will be accomplished. The model created in this study is expected to be precise enough to provide a jumping off point for the discovery of batik motifs in Central Java. A model built using a convolutional neural network (CNN) with the VGG16 and VGG19 architectures.

2. Literature Review

A different researcher carried out some previous research on batik classification. Research carried out by Alya *et al*. demonstrates that deep learning may be applied to categorizing batik motifs by utilizing transfer learning and CNN. In this research, the CNN model was combined with VGG16 neural network led to an accuracy of 89% for the test data [14]. Additional research was conducted to assist in identifying the motifs used in Indonesian batik. VGG16 and VGG19 are architectures currently used by CNNs. The study was conducted in three distinct settings: non-split datasets, split datasets, and rotated and scaled photographs. Studies comparing split to non-split datasets have demonstrated that the split dataset has higher accuracy, and VGG19 performs better on average by 1.1% compared to the non-split dataset. Experiments conducted with rotated images showed a 10% improvement in accuracy [15].

In a further line of investigation, a deep convolutional neural network and data augmentation were used to develop a software program for classifying the motifs used in batik textiles. ResNet18 and ResNet50 are the two CNN designs that are currently in use. Various types of augmentation data are utilized, such as rotation augmentation, random erase augmentation, scale augmentation, and flip augmentation. The accuracy of this research was 84.52% when using the ResNet18 architecture, while it was 81.90% when using the ResNet50 design. Increasing the use of augmentation data can improve accuracy by 8.52% for augmentation flip, 6.52% for augmentation scale, 9.38% for augmentation random erase, and 6.52% for augmentation rotation [16]. An additional line of research was taken to apply a convolutional neural network to categorize batik motifs. The CNN results have an average accuracy rate of approximately 65%. While the model produced by combining CNN and Grayscale may boost the results' average accuracy to 70%, it is essential to note that this is not guaranteed [17]. In a previous study, the CNN method and cross-validation accelerated by GPU CUDA were used to construct this model. The developed model has an accuracy of 90.14% on average and a time requirement of 29.56 seconds [18].

The five batik motifs that will comprise the dataset for this study are Banji, Kawung, Slope, Parang, and Slobog, all of which are from Central Java. In Sanskrit and Hindu-Buddhist regions, the Banji motif represents prosperity, longevity, and prosperous life. The basic shape of the Banji motif is a series of interconnected swastikas [19]. The Kawung pattern is derived from fro, a type of palm fruit. The motif is then deformed into an oval and organized in a cross representing the universe's structure; the cross is both a source of energy and a miniature universe [20]. The Slope motif has a diagonal angle of 45 degrees. One of the versions of the Slope motif that is well-known to the public is Udan Liris [19]. The Parang motif derives from the Javanese word 'Pereng,' which refers to the edge of a cliff that forms a diagonal line between the highlands and the lowlands [21]. Slobog is derived from the term Lobok, which signifies that those left behind can let go with an open and faithful heart [19]. Figure 1 depicts examples of batik motifs that will be utilized as datasets in this project.

Figure 1. Example Motif of Batik

3. Research Methodology

The process utilized to carry out this study comprises various stages, including (1) a literature review to gather information from books, journals, and other sources; (2) analysis of research-related issues encountered, and planned remedies; and (3) datasets of batik motifs gathered over the internet and directly retrieved using mobile devices by looking for batik motifs in Surakarta City. In addition, the next stages are: (4) Creating a model that is specific to the goals and outcomes of this study, (5) coding the model using the convolutional neural network technique after completing numerous steps such as dataset preparation, data preprocessing, and model training, (6) using data testing and an examination of the generated accuracy values. Then. model testing is a stage of model creation to be evaluated. This study employed a Visual Geometry Group (VGG) transfer learning model during the model-building phase.

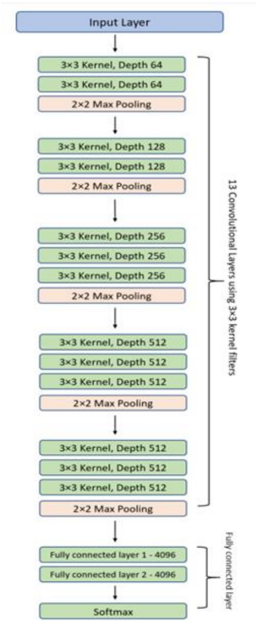

Figure 2. VGG-16 Architecture [23]

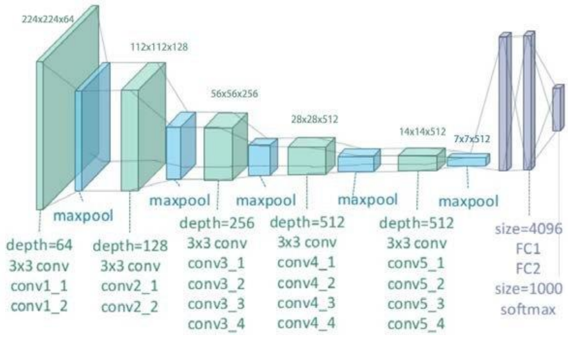

Figure 3. VGG-19 Architecture [25]

VGG is one of the CNN architectures that Karen Simonyan and Andrew Zisserman of Oxford University developed. This architecture competed in the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) by training on 14 million ImageNet images with 1000 classes and achieving a low classification error rate with an accuracy of 92.7% [22]. This VGG architecture's input dimensions are 224 by 224 pixels in RGB format. VGGNet is subdivided into numerous types based on the convolutional layers employed, with VGG16 and VGG19 being the most popular. VGG16 contains sixteen primary layers and a 3x3 kernel. The main layer consists of 13 convolutional layers, 2 fully connected layers, and 1 Softmax classifier [23]. VGG16 is the VGGNet architecture being challenged in ILSVRC. Figure 2 illustrates the architecture of VGG-16.

VGG19 is the result of the evolution of VGG16, the previous VGG architecture. VGG19 features 19 main layers and a 3x3 kernel, increasing from VGG16's 16 layers. The main layer comprises sixteen convolutional layers, two fully connected layers, and one Softmax classifier [24] [25]. Figure 3 displays the architecture of VGG-19.

4. Results and Discussion

4.1. Dataset

Five batik motifs from Central Java were selected for this study: Banji, Kawung, Slope, Parang, and Slobog. Each batik motif consists of 1,000 images or 200 photos. Retrieving datasets over the internet or in person is possible using a smartphone camera. Each photograph was collected in JPG format with RGB color depth. To improve the model's accuracy, each dataset will undergo data augmentation. The dataset will be separated into 80:15:5 training, validation, and testing groups.

4.2. Tools

In this study, Google Collaboratory as Cloud Computing with Intel(R) Xeon(R) CPU @ 2.30 GHz, 12 GB RAM, 164 GB Disk Space, and several auxiliary tools, including Python version 3.7.13, were utilized. As a model compiler, (2) Google Collaboratory employs a Tesla P100-PCIE-16GB GPU. (3) Recording live video on an iPhone 12 Pro.

4.3. Experiment Analysis

There will be six trials of the experiment. In the first, second, and third trials, the pre-trained VGG16 architecture model from Keras was utilized to construct models. In the fourth, fifth, and sixth trials, the pre-trained VGG19 architectural model from Keras was utilized to construct models. This research aimed to determine which of the two VGGNet topologies gave the most significant results. The first and fourth trials will use 50 epochs, the second and fifth will use 100 epochs, and the third and sixth will use 150 epochs (third and sixth experiments). There were five steps involved in conducting the study: (1) gathering datasets, (2) performing data augmentation and preprocessing on datasets gathered with the aid of Keras for the input of the model to be built, (3) creating models with VGG16 and VGG19 architectures using a pre-trained model from Keras that was trained with the ImageNet dataset and will be modified at the top layer, and (4) evaluating the performance of the designed models using the ImageNet dataset (classifier). The model is tested and evaluated by giving the confusion matrix results.

4.4. Result

The training and validation datasets are supplemented with new data using the ImageDataGenerator function of TensorFlow. This procedure is performed to increase the diversity of the used dataset. Next, the dataset is preprocessed to conform to model development specifications, which require an image of 224x224 pixels and an RGB picture format based on the VGGNet input size. This operation utilizes the Keras directory function's photo dataset. Transfer learning, a technique that employs a previously learned model with a dataset, is used to develop models. This research employed weights learned from the ImageNet dataset, which has over 14 million images and 1000 classifications, and the VGG16 and VGG19 neural network designs. Models constructed by setting a layer to non-trainable and then freezing it will utilize the framework model's convolutional foundation. The ImageNet-trained model weights will be retrieved and applied to the convolutional base. This ensures that the weight of the training layer remains consistent. While waiting for the model to be developed, the classifier layer will be removed from the model framework and replaced with a new classifier tailored to the model's requirements.

As a trainable layer, the classifier's weights can be modified as training progresses. The VGG19 architecture underwent a similar procedure. The classifier has four layers, starting with (1) Flatten, which converts many input dimensions into one dimension. (2) Dense activation ReLu, allowing for rapid network assembly. (3) Dropout, which picks input neuron units at random and ignores them during training to avoid overfitting. (4) Dense activation SoftMax makes understanding the activation SoftMax results as a probability distribution possible. There are more than two recognized items; thus, the classifier will be built using the Adam optimizer to reduce the error rate and apply the categorical cross-entropy loss. Every experiment was run with a different number of epochs.

The model will go through testing after it has completed the training procedure. Each dataset's actual and anticipated results will be shown using Matplotlib to illustrate the testing data. The dataset will be forecasted by using the model that has been constructed and saved in the prediction variable to use the prediction function. Each image's prediction label is kept in this variable. The tested dataset is then shown in the confusion matrix, and the model is then assessed using dataset testing in the evaluation function.

4.4.1. Experiment 1

The first experiment used the VGG16 architecture for 50 epochs. The results for training were a validation accuracy of 0.9333, validation loss of 0.6746, training accuracy of 0.8875, and training loss of 1.3546. The model's testing results in a testing accuracy of 0.9220 and a testing loss of 0.5814. The confusion matrix used to evaluate the model produces an overall accuracy of 0.9000 and a f1-score of 0.8997. Figures 4 and 5 show the results from experiment 1.

```
confusion matrix:
[[10  0  0  0  0]
 [ 0 10  0  0  0]
 [ 0  0  8  2  0]
 [ 0  0  3  7  0]
 [ 0  0  0  0 10]]
overall accuracy:  0.9000
```

	precisions	recall	f1-score	accuracy
Banji	1.0000	1.0000	1.0000	1.0000
Kawung	1.0000	1.0000	1.0000	1.0000
Lereng	0.8000	0.7273	0.7619	0.9000
Parang	0.7000	0.7778	0.7368	0.9000
Slobog	1.0000	1.0000	1.0000	1.0000
avg / total	0.9000	0.9010	0.8997	0.9600

Figure 4. Confusion Matrix Results, Experiment 1

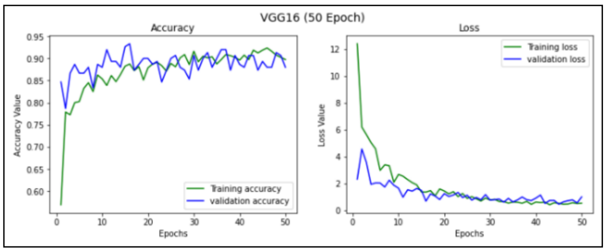

Figure 5. VGG16 Model Training Chart with 50 Epochs

4.4.2. Experiment 2

One hundred epochs of the VGG16 design were used in the second experiment. The final results for training were a validation accuracy of 0.94 and a loss of 0.7453, a training accuracy of 0.89, and a training loss of 0.6243. The model testing process produced results with a testing accuracy of 0.8998 and a testing loss of 0.5218. The confusion matrix results in an overall accuracy of 0.9000 and a f1-score of 0.8997 when testing the model. Figures 6 and 7 show the results from experiment 2.

```
confusion matrix:
[[10  0  0  0  0]
 [ 0 10  0  0  0]
 [ 0  0  7  3  0]
 [ 0  0  2  8  0]
 [ 0  0  0  0 10]]
overall accuracy:  0.9000
```

	precisions	recall	f1-score	accuracy
Banji	1.0000	1.0000	1.0000	1.0000
Kawung	1.0000	1.0000	1.0000	1.0000
Lereng	0.7000	0.7778	0.7368	0.9000
Parang	0.8000	0.7273	0.7619	0.9000
Slobog	1.0000	1.0000	1.0000	1.0000
avg / total	0.9000	0.9010	0.8997	0.9600

Figure 6. Confusion Matrix Results, Experiment 2

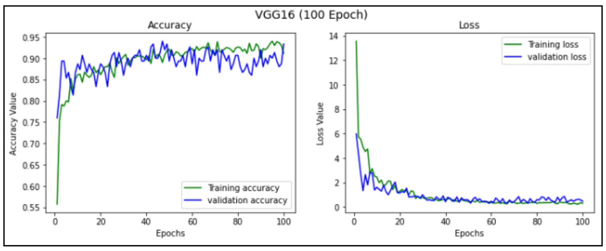

Figure 7. VGG16 Model Training Chart with 100 Epochs

4.4.3. Experiment 3

The third experiment used the VGG16 architecture throughout 150 epochs. The final training outcomes were 0.9467 accuracies in validation, 0.1371 loss in validation, 0.9287 accuracies in training, and 0.2732 loss in training. The model's testing results in a testing accuracy of 0.8799 and a testing loss of 0.4557. The confusion matrix used to test the model produces an overall accuracy of 0.88 and f1-score of 0.8792. Figures 8 and 9 show the results from experiment 3.

```
confusion matrix:
[[10  0  0  0  0]
 [ 0 10  0  0  0]
 [ 0  1  8  1  0]
 [ 0  0  3  7  0]
 [ 0  1  0  0  9]]
overall accuracy:  0.8800
```

	precisions	recall	f1-score	accuracy
Banji	1.0000	1.0000	1.0000	1.0000
Kawung	1.0000	0.8333	0.9091	0.9600
Lereng	0.8000	0.7273	0.7619	0.9000
Parang	0.7000	0.8750	0.7778	0.9200
Slobog	0.9000	1.0000	0.9474	0.9800
avg / total	0.8800	0.8871	0.8792	0.9520

Figure 8. Confusion Matrix Results, Experiment 3

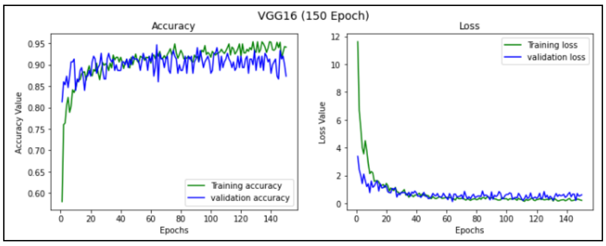

Figure 9. VGG16 Model Training Chart with 150 Epochs

4.4.4. Experiment 4

The fourth experiment used the VGG19 architecture for 50 epochs. The final training results were an accuracy of 0.8863, a loss of 0.5150, an accuracy of 0.9267, and a loss of 0.2231. The model's testing results in a testing accuracy of 0.8799 and a testing loss of 0.4967. The confusion matrix used to test the model produces an overall accuracy and f1-score of 0.9200 and 0.9198, respectively. Figures 10 and 11 show the results from experiment 4.

```
confusion matrix:
[[ 9  1  0  0  0]
 [ 0 10  0  0  0]
 [ 0  0  9  1  0]
 [ 0  0  2  8  0]
 [ 0  0  0  0 10]]
overall accuracy:  0.9200
```

	precisions	recall	f1-score	accuracy
Banji	0.9000	1.0000	0.9474	0.9800
Kawung	1.0000	0.9091	0.9524	0.9800
Lereng	0.9000	0.8182	0.8571	0.9400
Parang	0.8000	0.8889	0.8421	0.9400
Slobog	1.0000	1.0000	1.0000	1.0000
avg / total	0.9200	0.9232	0.9198	0.9680

Figure 10. Confusion Matrix Results, Experiment 4

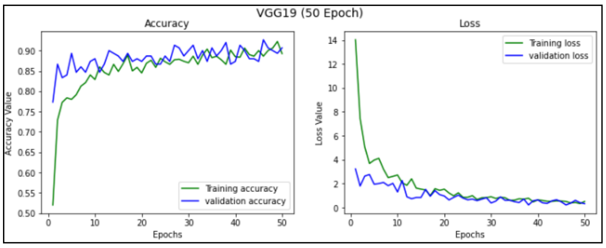

Figure 11. VGG19 Model Training Chart with 50 Epochs

4.4.5. Experiment 5

The VGG19 architecture was used in the sixth experiment, which included 100 epochs. The validation accuracy was 0.9333, the validation loss was 0.4484, and the validation accuracy was 0.8712. A testing loss of 0.5814 and a training loss of 0.9220 are obtained from the model's testing. The confusion matrix used to test the model produces an overall accuracy and f1-score of 0.96. Figures 12 and 13 show the results from experiment 5.

```
confusion matrix:
[[10  0  0  0  0]
 [ 0  9  0  1  0]
 [ 0  0  9  1  0]
 [ 0  0  3  7  0]
 [ 0  0  0  0 10]]
overall accuracy:  0.9000

             precisions    recall    f1-score    accuracy

     Banji      1.0000     1.0000     1.0000      1.0000

     Kawung     0.9000     1.0000     0.9474      0.9800

     Lereng     0.9000     0.7500     0.8182      0.9200

     Parang     0.7000     0.7778     0.7368      0.9000

     Slobog     1.0000     1.0000     1.0000      1.0000

     avg / total  0.9000   0.9056     0.9005      0.9600
```

Figure 12. Confusion Matrix Results, Experiment 5

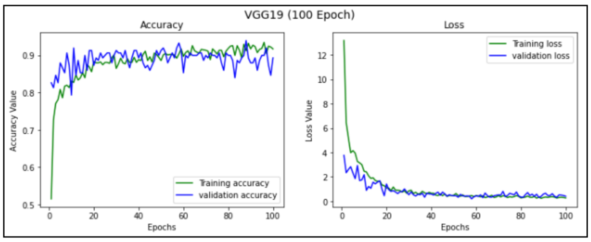

Figure 13. VGG19 Model Training Chart with 100 Epochs

4.4.6. Experiment 6

The VGG19 architecture was used in the sixth experiment, which included 150 epochs. The final training results were a training accuracy of 0.9100, training loss of 0.2705, validation accuracy of 0.9333, and validation loss of 0.2840. The model's testing results in a 0.4178 testing loss and a 0.8999 testing accuracy. When testing the model, the confusion matrix results in an overall accuracy of 0.94000 and a f1-score of 0.9398. Figure 14 shows the confusion matrix resulting from one of the trials to assess the model. Figure 15 shows the outcome of one test run while the model was being trained. The green line shows the training results, while the blue line represents the validation results. Table 1 shows the model training outcomes for the six experiments conducted, and Table 2 shows the model testing results.

```
confusion matrix:
[[10  0  0  0  0]
 [ 0 10  0  0  0]
 [ 0  0  8  2  0]
 [ 0  0  3  7  0]
 [ 0  0  0  0 10]]
overall accuracy:  0.9000

             precisions    recall    f1-score    accuracy

     Banji      1.0000     1.0000     1.0000      1.0000

     Kawung     1.0000     1.0000     1.0000      1.0000

     Lereng     0.8000     0.7273     0.7619      0.9000

     Parang     0.7000     0.7778     0.7368      0.9000

     Slobog     1.0000     1.0000     1.0000      1.0000

     avg / total  0.9000   0.9010     0.8997      0.9600
```

Figure 14. The Best Confusion Matrix Experiment Outcomes (VGG-16 with 50 epochs)

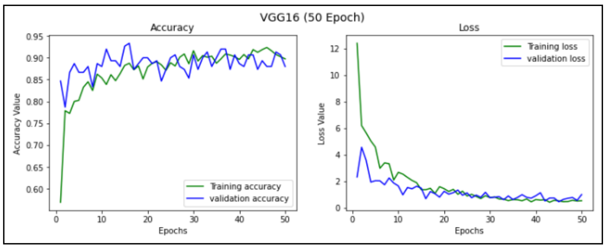

Figure 15. The Best Experiment Model Training Graph (VGG-16 with 50 epochs)

Table 1. Model Training Results in All Experiments

Experiment	Architecture	Number of Epochs	Training		Validation		Training Time (minutes)
			Acc	Loss	Acc	Loss	
1	VGG16	50	0.8875	1.3546	0.9333	0.6778	29.2815
2		100	0.8900	0.6243	0.9400	0.7453	56.5595
3		150	0.9287	0.2732	0.9467	0.1371	80.1191
4	VGG19	50	0.8863	0.5150	0.9267	0.2231	25.8198
5		100	0.9125	0.3417	0.9400	0.3174	58.9873
6		150	0.9100	0.2705	0.9333	0.2840	90.7449

Table 2. Model Testing Results in All Experiments

Experiment	Architecture	Number of Epochs	Test Loss	Test Acc	Overall Acc	F1-Score
1	VGG16	50	0.5814	0.9200	0.9000	0.8997
2		100	0.5218	0.8999	0.9000	0.8997
3		150	0.4557	0.8799	0.8800	0.8792
4	VGG19	50	0.4967	0.8799	0.9200	0.9198
5		100	0.4574	0.8799	0.9000	0.9005
6		150	0.4178	0.8999	0.9400	0.9398

The accuracy of the model testing process was better than 85%, and the accuracy of the model training process was greater than 90% based on the results of six tests. The architecture and quantity of epochs used to determine this. The number of epochs influences the training time. It will take longer if more epochs are employed, as shown in Table 1. The employed architecture, however, might also be impacted by the length of training. The training weights and layers mirror the architectural influence. The VGG16 design, however, requires more time to complete this experiment than the VGG19 architecture. This is a result of VGG19's inclusion of different settings and layers. VGG19 does not generate more accurate results than VGG16 because of the quantity of overfitting that takes place.

According to an analysis of the generated confusion matrix, the model tends to miss the Parang motif and be inconsistent with the Slope motif. This is because the Parang and Slope motifs share a similar fundamental shape. The Mlinjon that is used to divide the lines is what distinguishes the Parang and Slope motifs. Additionally, because Kawung motifs can be found as supplemental motifs in other patterns in some models, it might be challenging to identify between them.

The graph's changing behavior during model training indicates a rise or fall in accuracy. Additionally, the difference in accuracy values between training and validation is not seen as insignificant. Overfitting during the model-building process may be the cause of this. Each experiment incorporated the VGG-16, VGG-19, and epoch models. In the training model from the first experiment to the sixth experiment, overfitting frequently happens when VGG-19 is used. It is possible to detect overfitting by comparing the training loss to the validation loss. Experiments utilizing the VGG-19 model resulted in significant graph value alterations. The model with 150 iterations and VGG16 architecture performed the best during training in the third experiment, as shown in Table 1. The model has a 0.9287 training accuracy, a 0.2732 training loss, a 0.9467 validation accuracy, and a 0.1371 validation loss. Table 2 shows that the model with the VGG16 architecture and 50 epochs from the initial experiment is the best model for testing. The model's overall accuracy is 0.9000, testing accuracy is 0.9220, the testing loss is 0.5814, and f1-score is 0.8997.

The final model outputs are suboptimal since they result in significant loss values, suboptimal achieved accuracy (close to 100%), and variable accuracy levels across labels. There are several possible reasons for this: (1) The number of datasets employed is less varied in terms of the retrieved fragments and the pattern's shape. (2) The training, validation, and testing dataset's distribution ratio. (3) Inadequate data augmentation. (4) Creating a classifier layer that is not ideal.

5. Conclusions and Recommendations

A convolutional neural network (CNN) was successfully used to build a model to identify batik themes in Central Java, with the first, second, and third tests using the VGG16 architecture and the fourth, fifth, and sixth tests using the VGG19 architecture. The number of epochs used in each experiment varied. Five labels or classes are created from 1,000 RGB images, with training, validation, and testing dataset comparison ratios of 80%, 15%, and 5%, respectively. Out of the six trials, the model using the VGG16 architecture, and 150 epochs outperformed the training approach. The model's validation accuracy was 0.9467, and training accuracy was 0.9287. The model with 50 epochs and the VGG16 architecture performed the best testing procedure. The model achieves a f1-score on the confusion matrix of 0.8997, a testing accuracy of 0.9220, and an overall accuracy of 0.9000.

Some recommendations to advance the research include attempting to construct models using other CNN architectures, increasing the number of datasets, adding label variations, adding variations at the data augmentation stage, or applying grayscale at the data augmentation stage. For the best results, use a tuner to aid in the creation of a classifier layer from the built-in model.

Referensi

[1] Q. Yuan, S. Xu, & L. Jian, "A new method for retrieving batik shape patterns," *Journal of the Association for Information Science and Technology*, vol. 69, no. 4, pp. 578-599, 2018.
[2] "Arti kata batik - Kamus Besar Bahasa Indonesia (KBBI) Online." https://kbbi.web.id/batik (accessed Mar. 22, 2022).
[3] W. Rofianto, M. G. Haque, D. Sunarsi, A. Purwanto, & H. S. H. Sapari, "Cultural Product Branding, Antecedents, And Its Implications: A Study On The Context Of Indonesian Batik," *International Journal Of Social, Policy And Law*, vol. 2, no. 2, pp. 37-46, 2021.
[4] W. Steelyana, "Batik, a beautiful cultural heritage that preserve culture and support economic development in Indonesia," *Binus Business Review*, vol. 3, no. 1, pp. 116, 2012.
[5] R. A. Rahadi, D. Rahmawati, N. A. Windasari, & P. F. Belgiawan, "The Analysis of Consumers' Preferences for Batik Products in Indonesia," *Review of Integrative Business and Economics Research*, vol. 9, pp. 278-287, 2020.
[6] C. Y. Wang, "Building a network for preserving intangible cultural heritage through education: A study of Indonesian batik," *International Journal of Art & Design Education*, vol. 38, no. 2, pp. 398-415, 2019.
[7] S. I. S. Shaharuddin, M. S. Shamsuddin, M. H. Drahman, Z. Hasan, N. A. Mohd Asri, A. A. Nordin, & N. M. Shaffiar, "A review on the Malaysian and Indonesian batik production, challenges, and innovations in the 21st century," *SAGE Open*, vol. 11, no. 3, 2021.
[8] H. Widoyo, "Melestarikan budaya batik pada generasi muda," Feb. 02, 2021. https://binus.ac.id/character-building/2021/02/melestarikan-budaya-batik-pada-generasi-muda/(accessed Mar. 28, 2022).
[9] V. J. Basiroen, & J. O. Oetojo. (2018, November). Digital Campaign as a Solution to Engage the Millennial's Awareness about Javanese Batik Pesisir. In *1st International Conference* (vol. 276).
[10] A. N. Haniah, & P. F. Belgiawan, "Proposed Marketing Strategy for Increasing Batik Danar Hadi Purchase Intention among Gen Y and Z," *International Journal of Current Science Research and Review*, vol. 6, no. 1, pp. 229-237, 2023.
[11] T. Bariyah, M. A. Rasyidi, & Ngatini, "Convolutional Neural Network Untuk Metode Klasifikasi Multi-Label Pada Motif Batik," *Jurnal Teknologi Informasi Techno.Com*, vol. 20, no. 1, pp. 155–165, 2021.
[12] Pulung Adi Nugroho, I. Fenriana, & R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia," *JURNAL ALGOR*, vol. 2, no. 1, pp. 12–21, 2020.
[13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556
[14] R. F. Alya, M. Wibowo, & P. Paradise, "Classification of Batik Motif using Transfer Learning on Convolutional Neural Network (CNN)," *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 1, pp. 161-170, 2023.
[15] I. M. A. Agastya & A. Setyanto, "Batik Classification Utilizing Deep Learning Techniques and Data Augmentation," in Proceedings - 2018 3rd International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2018, Jul. 2018, pp. 27–31. doi: 10.1109/ICITISEE.2018.8720990.
[16] S. Febrian Tumewu, D. H. Setiabudi, & I. Sugiarto, "Klasifikasi Motif Batik menggunakan metode Deep Convolutional Neural Network dengan Data Augmentation," *Jurnal Infra*, vol. 8, no. 2, pp. 189–194, 2020.

[17] R. Mawan, “Klasifikasi motif batik menggunakan Convolutional Neural Network,” *JNANALOKA*, vol. 1, no. 1, pp. 45–50, Apr. 2020, doi: 10.36802/jnanaloka.2020.v1-no1-45-50.
[18] D. A. Mardani, *Deep Learning untuk Pengenalan Pola Batik Jawa*, Yogyakarta, Jul. 2019.
[19] A. Kusrianto, *Batik Filosofi, Motif dan Kegunaan*, Penerbit Andi, 2013.
[20] A. N. Syafitri, *Tanaman Bunga Telang Dan Kawung Sebagai Inspirasi Penciptaan Busana Pesta Muslim* (Doctoral dissertation, Institut Seni Indonesia Yogyakarta), 2023.
[21] N. W. Kartika, *LKP: Studi Kasus pada Yayasan Sekolah Ekspor Nasional: Perencanaan Ekspor ke Uni Emirat Arab pada UMKM Eunoia Bateek* (Doctoral dissertation, Universitas Dinamika), 2023.
[22] I. B. K. Sudiatmika & I. G. A. A. S. Dewi, “Pengenalan Karakter Wayang Bali dengan Arsitektur VGG-16,” *Jurnal Teknologi Informasi dan Komputer (JUTIK)*, vol. 7, no. 1, pp. 47–53, 2021.
[23] S. Tammina, “Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images,” *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, pp. 143–150, 2019.
[24] C. Yulian, “Penerapan Deep Learning Style Transfer untuk Menciptakan Kombinasi Gambar Digital dengan Batik,” Universitas Multimedia Nusantara, Tangerang, 2021.
[25] I. Chebbi, “VGG19 For VQGR: Visual Generation of Relevant Natural Language Questions from Radiology Images for Anomaly Detection,” in Conference and Labs of the Evaluation Forum, September 21–24, 2021.