Image Classification of *Wayang* Using Transfer Learning and Fine-Tuning of CNN Models

Muhammad Banjaransari, Adhi Prahara
Informatics Department, Universitas Ahmad Dahlan, Indonesia

ARTICLE INFORMATION

Article History:

Submitted 25 December 2023
Revised 04 February 2024
Accepted 07 February 2024

Keywords:

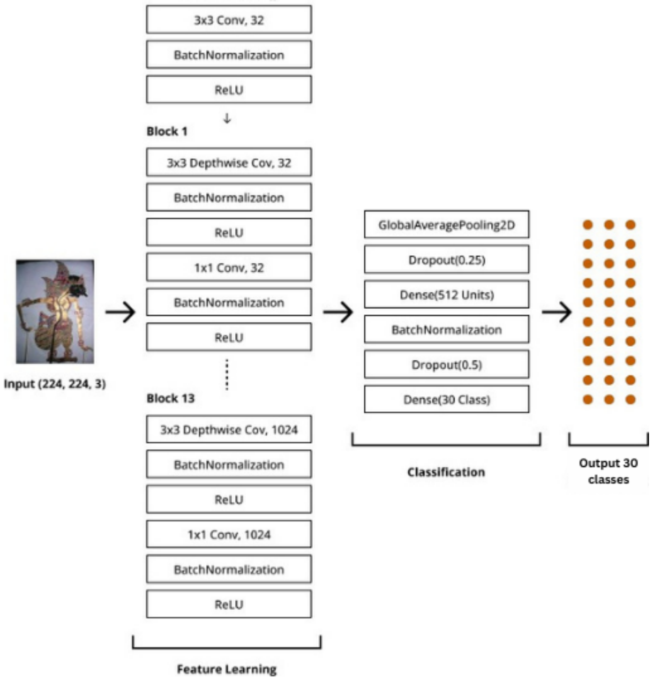
Wayang;
Transfer Learning;
CNN;
Classification;
Computer Vision

Corresponding Author:

Adhi Prahara,
Universitas Ahmad Dahlan,
Yogyakarta, Indonesia.
Email:
adhi.prahara@tif.uad.ac.id

This work is licensed under a Creative Commons Attribution-Share Alike 4.0

ABSTRACT



Wayang (shadow puppetry) is a traditional puppetry used in a performance to tell a story about the heroism of its main characters. *Wayang* has gained recognition as a cultural masterpiece by UNESCO. However, this cultural heritage now declining and not many people know about *wayang*. One of the solutions is using computer vision technology to classify *wayang* images. In this research, a transfer learning approach using Convolutional Neural Network (CNN) models namely MobileNetV2 and VGG16 followed by fine-tuning was proposed to classify *wayang*. The dataset consists of 3,000 images divided into 30 classes. This data is split into training and test data that are utilized for training and evaluating the model. Based on the evaluation, the MobileNetV2 model achieved precision, recall, F1-score, and accuracy of 95%, 94%, 94%, and 94.17%, respectively. Meanwhile, the VGG-16 model obtained 93% for all metrics. It can be concluded that transfer learning and fine-tuning using the MobileNetV2 model produces the best result in classifying *wayang* images compared to the VGG16 model. With good performance, the proposed method can be implemented on mobile applications to provide information about *wayang* from the captured images, thus indirectly supporting the preservation of cultural heritage in Indonesia.

Document Citation:

M. Banjaransari and A. Prahara, “Image Classification of Wayang Using Transfer Learning and Fine-Tuning of CNN Models,” *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 5, no. 4, pp. 632-641, 2023, DOI: 10.12928/biste.v5i4.9977.

1. INTRODUCTION

Wayang (shadow puppetry) is a traditional puppetry used in a performance to tell a story of heroic characters who fight villainous characters. *Wayang* records various historical events from generation to generation and become a culture, particularly in Java and Bali. The *wayang* known today is a cultural heritage that is believed to have existed for approximately 1,500 years [1]. *Wayang* has gained recognition as a cultural masterpiece by UNESCO. Since UNESCO established a convention in 1972 related to the cultural heritage of objects, locations, and natural landscapes, there has been an understanding that intangible cultural heritage and oral traditions should also be preserved. This understanding is based on the notion that cultural heritage, rich in values and meanings, is endangered by globalization or environmental destruction. The artistic value of *wayang* also has undergone some changes. *Wayang* performances are considered outdated and less appealing. Another factor that contributes to the decline of *wayang* in society is the lack of understanding of *wayang* stories, especially the characters. Because there are numerous characters in a story such as Ramayana and Mahabharata [2], it is difficult to remember all of the names and the characteristics of each *wayang*. This issue is likely to worsen for future generations [3]. Therefore, some efforts were made to introduce *wayang* to the younger generation for example using computer vision technology that can classify objects from the images captured by camera or smartphone.

Image classification is used to group images into specific categories. The idea behind image classification is to provide the computer with a series of features extracted from the image, train the features using a model, and return a value indicating the category of the image. This method can also be used to classify *wayang* from an image [4]–[12]. Nurhikmat uses a Convolutional Neural Network (CNN) model to recognize *wayang golek* (wooden puppetry) from images to address societal issues of forgetting traditional culture. The performance evaluation shows that the CNN model achieves a training accuracy of 95% and a validation accuracy of 90%, The trained model is tested on new data, achieving an accuracy of 93% [4]. Yudianto *et al.* apply the CNN model to recognize *wayang punakawan* from images to address the declining interest in *wayang*. This research achieves the best accuracy of 97%, precision of 93%, and recall of 87% [5]. Sudiatmika and Dewi use a CNN model namely VGG16 to recognize Balinese *wayang* from images. The training process is evaluated on several epochs, such as 50, 100, 150, and 200. The best results are achieved in the 200th epoch with an accuracy of 89% [6]. Training a deep learning model requires a large dataset to achieve high performance. With a limited number of wayang datasets, training the model from scratch is not efficient.

This research proposed a classification of *wayang* using transfer learning and fine-tuning of CNN models. The contributions of this work are shown in the use of transfer learning and fine-tuning to classify *wayang* images from limited datasets and to support the preservation of cultural heritage by providing a solution to introduce *wayang* characters. The rest of this paper is organized as follows. Section 2 presents the proposed method, Section 3 shows and discusses the result, and the conclusion of this work is presented in Section 4.

2. METHODS

A flowchart of the proposed *wayang* classification using transfer learning of CNN models is shown in Figure 1. The method consists of data collection, preprocessing, building the model, transfer learning from the pre-trained CNN models, fine-tuning, and performance evaluation. The details of each step are described in the following sub-section.

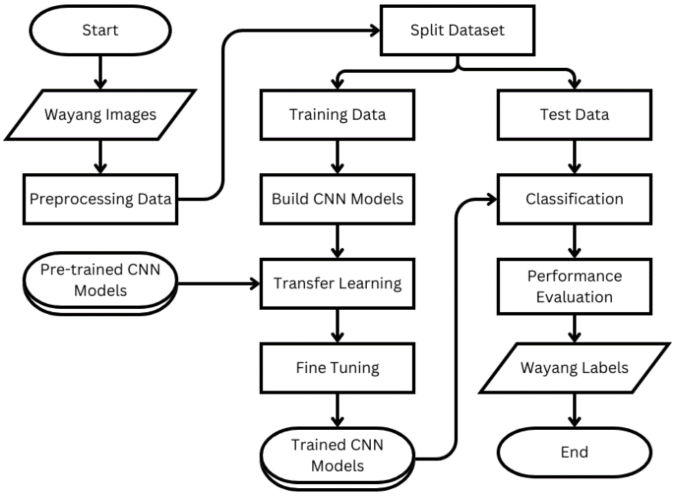


Figure 1. The proposed *wayang* classification using transfer learning of CNN models

2.1. Data Collection

In this research, *wayang* dataset was obtained mainly from Kaggle [13] and Google Image Search. The dataset has various backgrounds and lighting conditions. The dataset consists of 3,000 *wayang* images which are divided into 100 images per class to avoid class imbalance because it may affect the performance of the deep learning model [14]. The classes are *wayang* characters namely *Abimanyu, Adipati Karna, Antasena, Arjuna, Aswatama, Baladewa, Banowati, Basudewa, Bima, Bisma, Burisrawa, Cakil, Dropadi, Dursasana, Duryudana, Gatot Kaca, Hanoman, Kresna, Kunti, Nakula, Narada, Setyaki, Srikandi, Yudhistira, Anggada, Kumbakarna, Semar, Gareng, Petruk*, and *Bagong*. They are the most important characters in the Ramayana and Mahabharata story [2].

2.2. Preprocessing Data

Preprocessing is used to prepare the data before training. Preprocessing steps such as resizing with padding, dataset splitting, data augmentation, and standardization of input are performed. Resizing with padding is useful to keep the aspect ratio of *wayang*. The dataset is divided into training and test data with a ratio of 4:1. The training data is split to generate validation data for training. Data augmentation will enrich the data because wayang images may be captured from different angles, lighting conditions, and distances. Standardization of input data is also important in deep learning models [15]. The final preparation is to create batches of images from the dataset.

2.3. CNN Models

Convolutional Neural Network (CNN) [16] is a popular model for dealing with image data. It consists of convolutional layers where the input images are processed using specific filters. Each layer produces pattern formations from various parts of the image, making it easier for categorization. The pooling layer reduces the dimension of a matrix by applying pooling operations. The pooling layer is generally placed after the convolutional layer. Two common types of pooling layers are average pooling and max pooling. The fully connected layer is used to transform data dimensions for classification. This research uses two pre-trained models of CNN namely VGG16 [17] and MobileNetV2 [18]. The pre-trained model from ImageNet [19] is available on Tensorflow [20].

MobileNetV2 is one of the CNN architectures developed specifically for image recognition tasks on mobile devices with limited computational resources. The MobileNetV2 architecture employs a technique called depthwise separable convolution to reduce the number of parameters and computational operations, making it more computationally efficient. To perform transfer learning, the classification part of MobileNetV2 is replaced to fit the number of output categories/classes (see Figure 2).

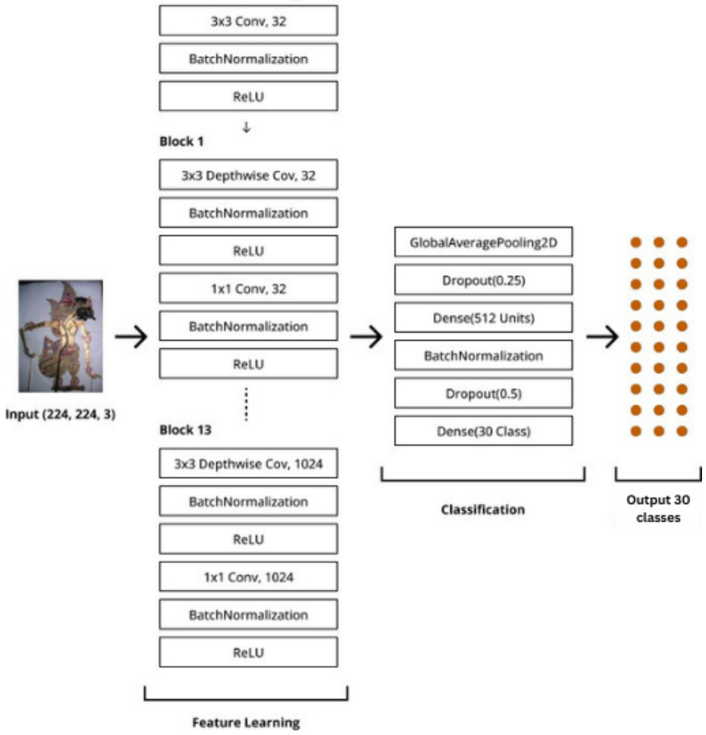


Figure 2. Transfer learning using the MobileNetV2 model

VGG16 model takes input images with a size of 224x224 pixels. The convolutional layers share the same parameters, using filters with a size of 3x3 and a stride of 1x1. VGG16 model consists of a total of 16 layers, with 13 feature extraction layers and 3 fully connected layers. To perform transfer learning, the classification part of VGG16 is replaced to fit the number of output categories/classes (see Figure 3). In both models, dropout and batch normalization are used to achieve generalization.

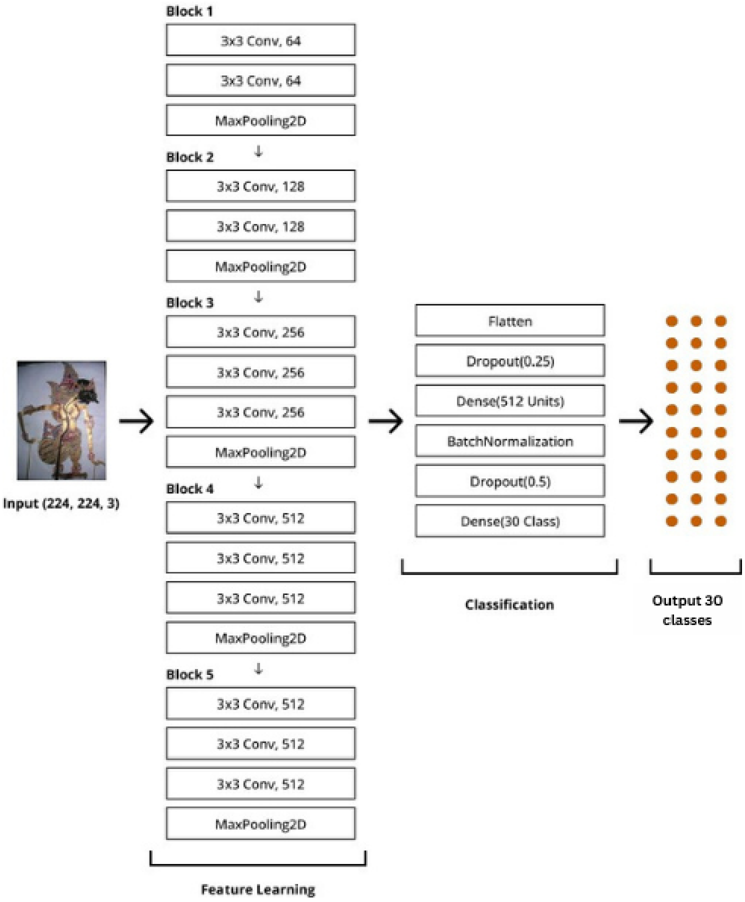


Figure 3. Transfer learning using the VGG16 model

2.4. Transfer Learning and Fine-Tuning

Transfer learning uses a model that has been trained on one task or dataset to speed up the learning process on a different, related, or similar task or dataset. By leveraging the knowledge acquired by the model from a previous source, transfer learning can help overcome data limitations and accelerate convergence when training a model on a new task [21]. In short, transfer learning involves taking features learned from one problem and applying them to a new or similar problem. Transfer learning utilizes pre-trained models that are already trained on large datasets such as ImageNet [22]. In the training process, all of the feature extraction layers are frozen and only the classification layers are trained because the base convolutional layers from the pre-trained model are already able to extract useful features for image classification. After transfer learning, the parameters of the model are adjusted using fine-tuning method. All of the feature extraction layers are unfroze and included in the training. This method will adjust the feature representation from the base model to fit the dataset. The process is done with a smaller learning rate to increase the accuracy of the model.

2.5. Evaluation Metrics

The performance of a multi-class classification model can be assessed through some metrics namely accuracy, recall, precision, and F1-score [23]. To calculate these metrics, a matrix known as the confusion matrix is required. The equations of accuracy, precision, recall, and F1-score are shown in Equations (1), (2), (3), and (4) respectively where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\ Score = 2\left(\frac{precision\ \cdot\ recall}{precision + recall}\right) \tag{4}$$

3. RESULT AND DISCUSSION

The proposed method is implemented using Python with additional libraries such as Tensorflow, OpenCV, and Scikit-learn. The program runs on a cloud computer from Google Colab with T4 GPU to speed up the training.

3.1. *Wayang* Dataset

The number of *wayang* in the dataset is 30 characters with 100 images per character. The total number of images is 3.000 which is further divided into 2.400 training images and 600 test images. The sample of *wayang* dataset is shown in Figure 4. Based on Figure 4, the dataset has various background and lighting conditions.

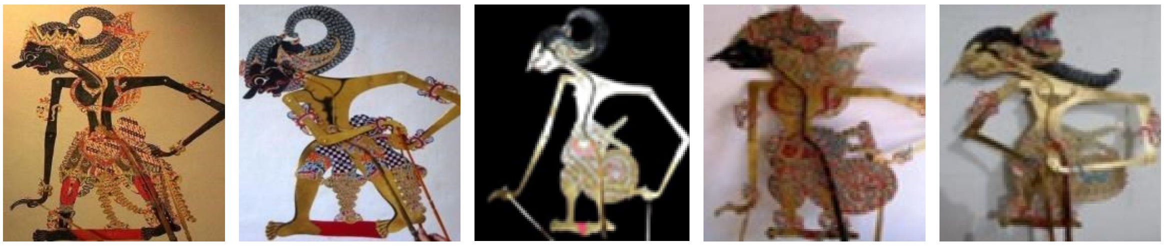


Figure 4. Sample of *wayang* dataset

3.2. Preprocessing Result

Preprocessing is used to prepare the dataset before training. The results of each step in the preprocessing data are shown as follows.

1) Resize with padding is used to maintain the aspect ratio of the image by adding additional pixels to the image border. Figure 5 shows the original image with zero padding to maintain its aspect ratio when resized to the input size of 224x224x3.

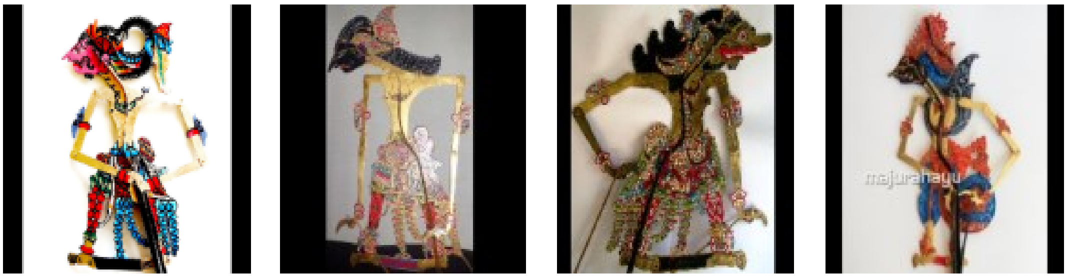


Figure 5. The result of resizing with zero padding on *wayang* image

2) Data augmentation is a process that involves applying various transformations to images in the dataset. The main goal is to increase the quantity and variability in the dataset. Data augmentation is performed using operations such as rotation with a range of 20 degrees and horizontal flipping. The result is shown in Figure 6 where the image on the left is the original image and the rest is the result of image augmentation. The total number of images produced after augmentation is 15.000 *wayang* images.

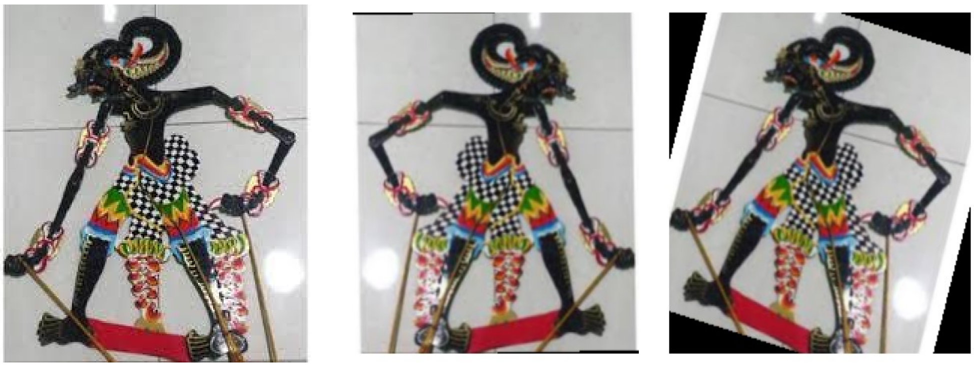


Figure 6. The result of augmentation on *wayang* image

3) All pixels in the input images are standardized to a range of [0, 1] for VGG16 and [-1, 1] for MobileNetV2 to match the configuration of the pre-trained models.

3.3. Transfer Learning Result

The training process uses Adam as an optimizer, a learning rate of 0.001, and 150 epochs. The final trained model is chosen based on the lowest validation loss during training. Transfer learning using the MobileNetV2 model achieves a training accuracy of 87.39% and a validation accuracy of 82.67%. Meanwhile, the evaluation results on the test data to assess the model's performance achieved an accuracy of 85.33%. The training graph for the MobileNetV2 model can be seen in Figure 7. Based on the training graph, the model is slightly overfitting although a kernel regularization was added in the layers.

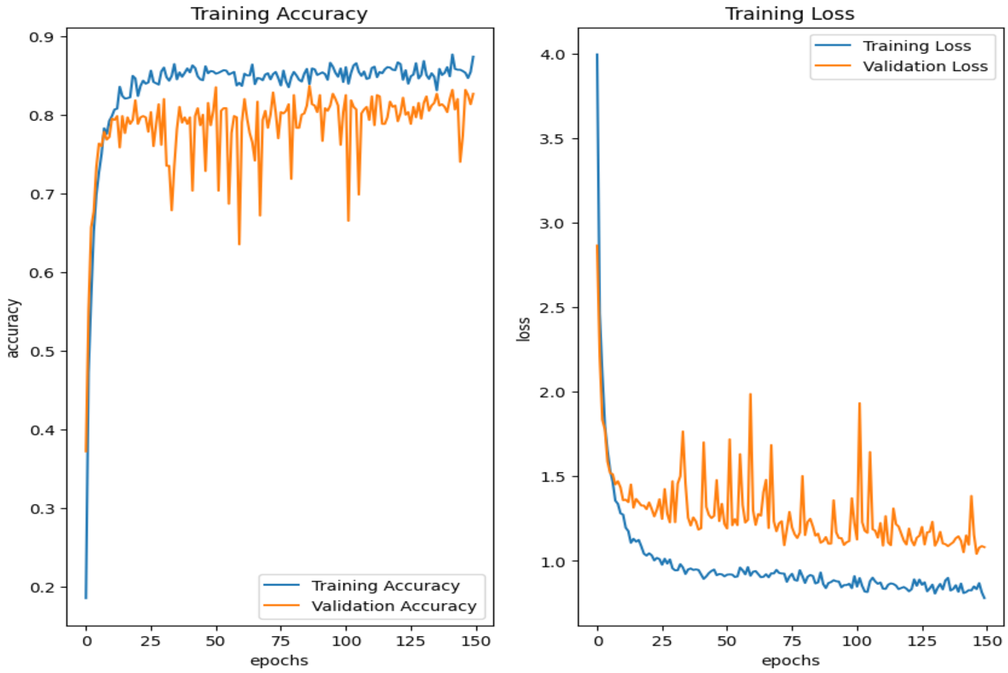


Figure 7. The training graph of transfer learning MobileNetV2 model

Transfer learning using the VGG16 model achieves a training accuracy of 90.33% and a validation accuracy of 82.83%. The evaluation results on the test data obtained an accuracy of 83%. The training graph for the VGG16 model can be seen in Figure 8. Although the validation graph is not stable, the training process is set to save the best model according to the lowest validation loss.

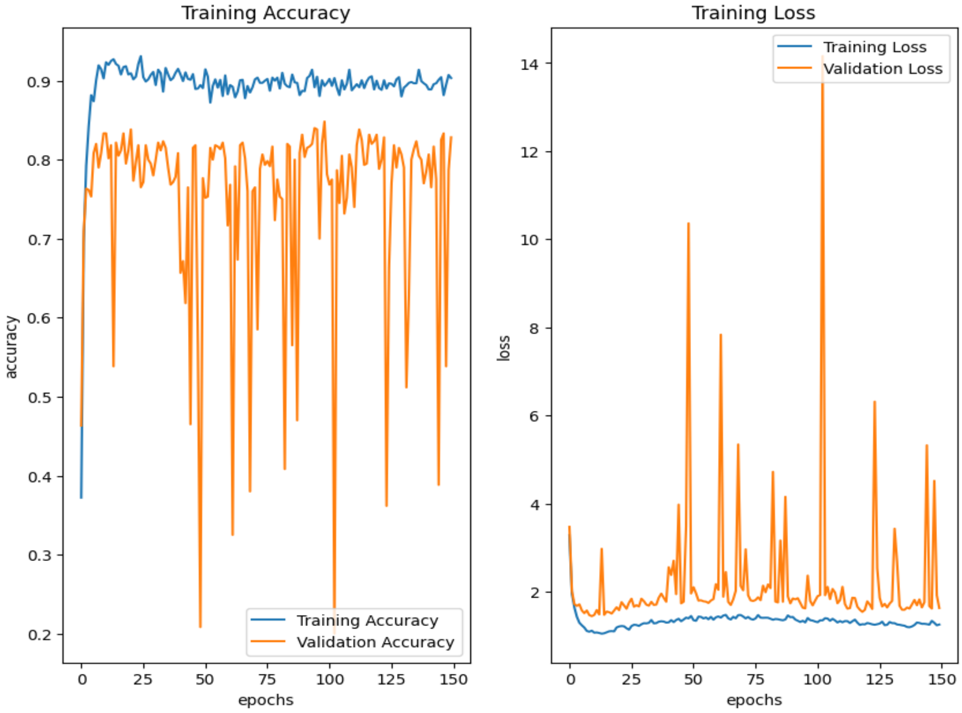


Figure 8. The training graph of transfer learning VGG16 model

3.4. Fine Tuning Result

Based on the training graph of transfer learning using CNN models in Figure 7 and Figure 8, the model has difficulty learning the new data. This is the reason why fine-tuning is an important step after transfer learning. Fine-tuning also trains the feature extraction layers and slowly adjusts the weights to fit the new data. The hyperparameter configuration is similar to transfer learning except the learning rate for fine-tuning is set to 0.0001. The fine-tuning process on the MobileNetV2 model achieves a training accuracy of 96.11% and a validation accuracy of 89.17%. Meanwhile, the evaluation results on the test data achieved an accuracy of 94.17%. The training graph from the result of fine-tuning the MobileNetV2 model can be seen in Figure 9.

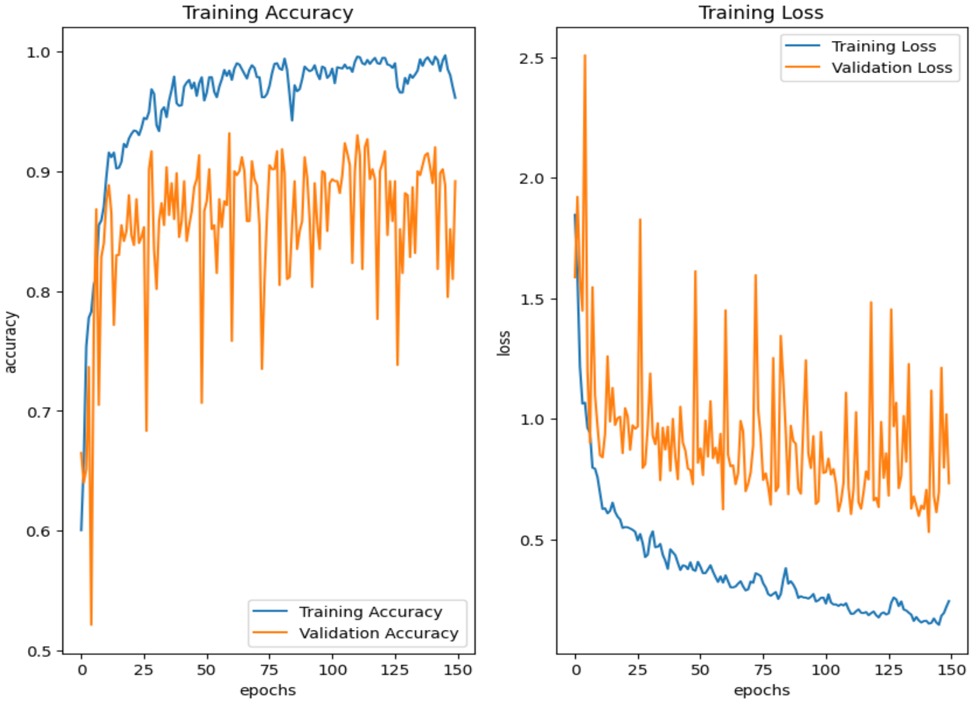


Figure 9. Training graph from the result of fine-tuning the MobileNetV2 model

The fine-tuning process on the VGG16 model yielded a training accuracy of 99.72% and a validation accuracy of 90.67%. The evaluation results on the test data obtained an accuracy of 93%. The training graph from the result of fine-tuning the VGG16 model can be seen in Figure 10. The fluctuating validation graph is caused by various factors such as difficult-to-predict dataset variations, and model hyperparameters like learning rate and regularization, such as dropout.

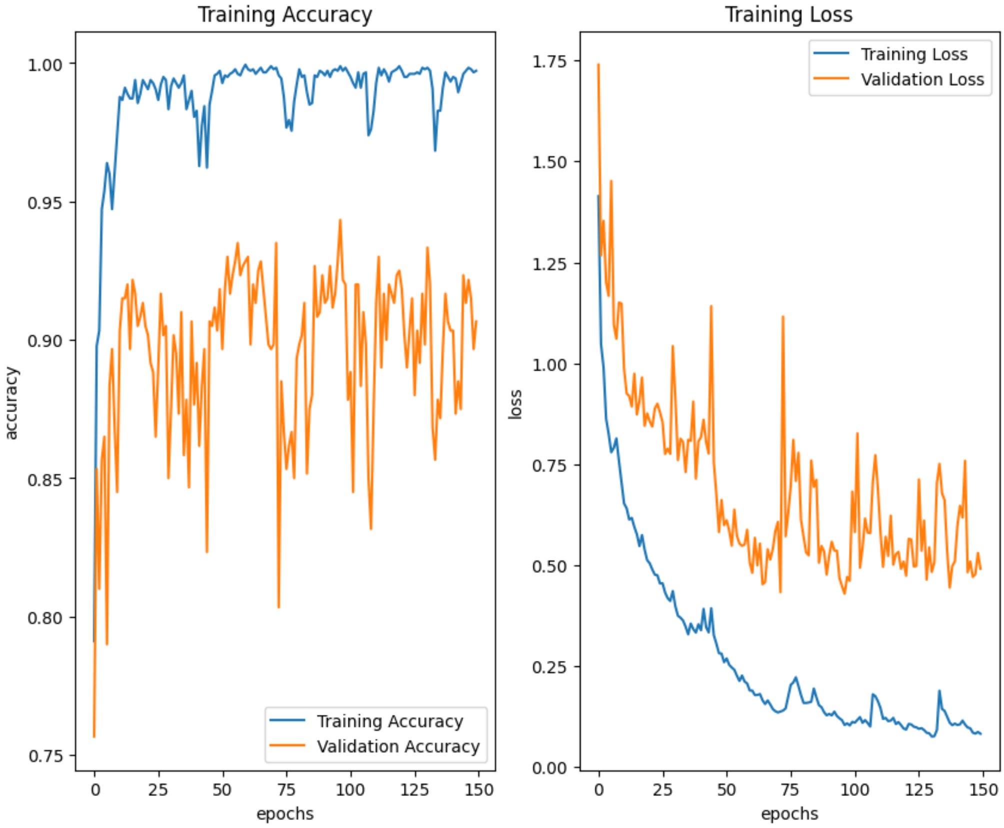


Figure 10. Training graph from the result of fine-tuning the VGG16 model

3.5. Performance Evaluation

The model was evaluated using test data consisting of 600 images. The confusion matrix was employed to assess the model's performance in classifying *wayang* images (see Table 1). Based on Table 1, the lowest performance from the MobileNetV2 model is on the *Nakula* class with 75% accuracy. Some images may have poor quality and inevitably affect the test results. The total correctly predicted data out of the test dataset was 565 images out of 600. The accuracy achieved using the MobileNetV2 model is 94.17%. For the VGG16 model, the lowest performance is on the *Nakula* class with 70% accuracy. The total correctly predicted data out of the test dataset was 558 images out of 600. The accuracy achieved using the VGG16 model is 93%.

The performance of both models is compared to determine the best model for the *wayang* classification task. Table 2 shows the result of training, validation, and test accuracy for both models. Based on Table 2, the highest accuracy on test data is achieved by the MobileNetV2 model although the VGG16 model scored higher in the training process. The fine-tuning process also increases the accuracy significantly.

Based on the confusion matrix, some metrics such as precision, recall, F1-score, and accuracy are computed. Table 3 shows the performance evaluation result of both models. The MobileNetV2 model achieved precision, recall, F1-score, and accuracy of 95%, 94%, 94%, and 94.17%, respectively. Meanwhile, the VGG-16 model obtained 93% for precision, recall, F1-score, and accuracy. From these results, it can be concluded that the MobileNetV2 model produces the best evaluation in classifying *wayang* images compared to the VGG16 model.

Table 1. Comparison of prediction result

No	Class Name	Number of Data	MobileNetV2 Prediction Result	VGG16 Prediction Result
1	*Abimanyu*	20	100%	95%
2	*Adipati Karna*	20	95%	90%
3	*Anggada*	20	100%	95%
4	*Antasena*	20	100%	100%
5	*Arjuna*	20	95%	85%
6	*Aswatama*	20	95%	95%
7	*Bagong*	20	95%	100%
8	*Baladewa*	20	100%	95%
9	*Banowati*	20	90%	85%
10	*Basudewa*	20	85%	90%
11	*Bima*	20	85%	85%
12	*Bisma*	20	90%	90%
13	*Burisrawa*	20	95%	95%
14	*Cakil*	20	95%	95%
15	*Drupadi*	20	100%	100%
16	*Dursasana*	20	85%	95%
17	*Duryudana*	20	100%	100%
18	*Gareng*	20	90%	90%
19	*Gatot Kaca*	20	85%	90%
20	*Hanoman*	20	100%	100%
21	*Kresna*	20	100%	95%
22	*Kumbakarna*	20	100%	95%
23	*Kunti*	20	95%	95%
24	*Nakula*	20	75%	70%
25	*Narada*	20	100%	100%
26	*Petruk*	20	100%	100%
27	*Semar*	20	100%	95%
28	*Setyaki*	20	90%	100%
29	*Srikandi*	20	85%	75%
30	*Yudhistira*	20	100%	95%
	Total	600	94.17%	93%

Table 2. Comparison accuracy during training and testing

Model	Step	Epoch	Training Accuracy	Validation Accuracy	Test Accuracy
MobileNetV2	Transfer learning	150	87,39%	82,67%	85,33%
	Fine-tuning	150	96,11%	89,17%	94,17%
VGG16	Transfer learning	150	90,33%	82,83%	83%
	Fine-tuning	150	99,72%	90,67%	93%

Table 3. Comparison of performance evaluation

Model	Precision	Recall	F1-Score	Accuracy
MobileNetV2	95%	94%	94%	94,17%
VGG-16	93%	93%	93%	93%

4. CONCLUSIONS

Based on the results, it can be concluded that the proposed method is capable of classifying *wayang* from images. The proposed method uses CNN models, namely MobileNetV2 and VGG16, through the application of transfer learning and fine-tuning methods. The dataset consists of 3,000 images from 30 classes, which were divided into 2,400 training data and 600 test data. The training results for the MobileNetV2 model after fine-tuning achieved an accuracy of 96.11%, while the training results for the VGG-16 model after fine-tuning reached an accuracy of 99.72%. However, based on the performance of test data, the MobileNetV2 model achieved precision, recall, F1-score, and accuracy of 95%, 94%, 94%, and 94.17%, respectively. Meanwhile, the VGG-16 model obtained 93% for precision, recall, F1-score, and accuracy. From these results, it can be concluded that the MobileNetV2 model provides better evaluation in classifying *wayang* images compared to the VGG16 model. With good performance evaluation, the proposed method can be implemented on mobile applications to provide information about wayang from the captured images, thus indirectly supporting the preservation of cultural heritage in Indonesia.

REFERENCES

[1] B. H. Prilosadoso, B. Pujiono, S. Supeni, and B. W. Setyawan, "Wayang beber animation media as an effort for preserving wayang tradition based on information and technology," In *Journal of Physics: Conference Series*, vol. 1339, no. 1, p. 012109, 2019, https://doi.org/10.1088/1742-6596/1339/1/012109.

[2] J. C. Oman. *The Stories of the Ramayana and the Mahabharata*. Routledge. 2019. https://books.google.co.id/books?hl=id&lr=&id=0I2lDwAAQBAJ.

[3] R. C. I. Prahmana and A. Istiandaru, "Learning sets theory using shadow puppet: A study of Javanese ethnomathematics," *Mathematics*, vol. 9, no. 22, p. 2938. 2021, https://doi.org/10.3390/math9222938.

[4] D. P. Prabowo, M. K. A. Nugraha, D. I. I. Ulumuddin, R. A. Pramunendar and S. Santosa, "Indonesian Traditional Shadow Puppet Classification using Convolutional Neural Network," *2021 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pp. 1-5, 2021, https://doi.org/10.1109/iSemantic52711.2021.9573218.

[5] M. R. A. Yudianto and H. Al Fatta, "The effect of Gaussian filter and data preprocessing on the classification of Punakawan puppet images with the convolutional neural network algorithm," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 12, no. 4, 2022, https://doi.org/10.11591/ijece.v12i4.pp3752-3761.

[6] W. Supriyanti and D. A. Anggoro, "Classification of Pandavas Figure in Shadow Puppet Images using Convolutional Neural Networks," *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, vol. 7, no. 1, 2021, https://doi.org/10.23917/khif.v7i1.12484.

[7] A. Mustafid, M. M. Pamuji, and S. Helmiyah, "A Comparative Study of Transfer Learning and Fine-Tuning Method on Deep Learning Models for Wayang Dataset Classification," *IJID (International J. Informatics Dev.*, vol. 9, no. 2, pp. 100–110, 2020, https://doi.org/10.14421/ijid.2020.09207.

[8] A. P. Wibawa, W. A. Y. Pratama, A. N. Handayani, and A. Ghosh, "Convolutional Neural Network (CNN) to determine the character of wayang kulit," *Int. J. Vis. Perform. Arts*, vol. 3, no. 1, pp. 1–8, 2021, https://doi.org/10.31763/viperarts.v3i1.373.

[9] I. B. K. Sudiatmika, M. Artana, N. W. Utami, M. A. P. Putra, and E. G. A. Dewi, "Mask R-CNN for indonesian shadow puppet recognition and classification," in *Journal of Physics: Conference Series*, 2021, vol. 1783, no. 1, p. 12032, https://doi.org/10.1088/1742-6596/1783/1/012032.

[10] Muhathir and Al-Khowarizmi, "Measuring the Accuracy of SVM with Varying Kernel Function for Classification of Indonesian Wayang on Images," *2020 International Conference on Decision Aid Sciences and Application (DASA)*, pp. 1190-1196, 2020, https://doi.org/10.1109/DASA51403.2020.9317197.

[11] M. R. Arif Yudianto, R. A. Hasani, and P. Sukmasetya, "Study comparison of color channel to median filter in wayang image using convolutional neural network algorithm," in *AIP Conference Proceedings*, vol. 2706, no. 1, 2023, https://doi.org/10.1063/5.0120384.

[12] N. Khairina, R. K. I. Barus, M. Ula, I. Sahputra, and others, "Preserving Cultural Heritage Through AI: Developing LeNet Architecture for Wayang Image Classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 9, 2023, https://doi.org/10.14569/IJACSA.2023.0140919.

[13] Lis, M. (2019). The Art of Wayang. *Art of the Orient*, *8*, 137-150, 2019, https://doi.org/10.15804/aoto201907.

[14] K. Ghosh, C. Bellinger, R. Corizzo, P. Branco, B. Krawczyk, and N. Japkowicz, "The class imbalance problem in deep learning," *Mach. Learn.*, pp. 1–57, 2022, https://doi.org/10.1007/s10994-022-06268-8.

[15] J. Folmsbee, S. Johnson, X. Liu, M. Brandwein-Weber, and S. Doyle, "Fragile neural networks: the importance of image standardization for deep learning in digital pathology," in *Medical Imaging 2019: Digital Pathology*, vol. 10956, pp. 222–228, 2019, https://doi.org/10.1117/12.2512992.

[16] J. D. Kelleher. *Deep learning*. MIT press. 2019. https://books.google.co.id/books?hl=id&lr=&id=b06qDwAAQBAJ.

[17] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM," *Sensors*, vol. 21, no. 8, p. 2852, 2021, https://doi.org/10.3390/s21082852.

[18] S. R. Joshi and M. El-Sharkawy, "HBONext: HBONet with Flipped Inverted Residual," *2021 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS)*, pp. 1-5, 2021, https://doi.org/10.1109/DTS52014.2021.9498121.

[19] Z. Zhu, *et al.*, "Webface260m: A benchmark unveiling the power of million-scale deep face recognition," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10492-10502, 2021, https://doi.org/10.1109/CVPR46437.2021.01035.

[20] N. Kumar, M. Rathee, N. Chandran, D. Gupta, A. Rastogi and R. Sharma, "CrypTFlow: Secure TensorFlow Inference," *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 336-353, 2020, https://doi.org/10.1109/SP40000.2020.00092.

[21] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning--ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*, pp. 270–279, 2018, https://doi.org/10.1007/978-3-030-01424-7_27.

[22] G. Vrbančič and V. Podgorelec, "Transfer Learning With Adaptive Fine-Tuning," in *IEEE Access*, vol. 8, pp. 196197-196211, 2020, https://doi.org/10.1109/ACCESS.2020.3034343.

[23] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," *arXiv Prepr. arXiv2008.05756*, 2020, https://doi.org/10.48550/arXiv.2008.05756.