Recognition of the Lima Pandawa Shadow Puppet characters utilizing Principal Component Analysis (PCA) for feature extraction and K-Nearest Neighbor (KNN) for classification

I Widaningrum*[1], I P Astuti[2], D Mustikasari[3], K Nurfitri[4], R R Az-Zahra[5], R I Vidyastari[6], A Selamat[7]

[1-6]Universitas Muhammadiyah Ponorogo

[7]Universiti Teknologi Malaysia Malaysia

E-mail: iwidaningrum@umpo.ac.id[1], indahsan.0912@gmail.com[2], dyahmustika@umpo.ac.id[3], nurfitrikhoiru9@gmail.com[4], rifqizahra31@gmail.com[5], rhesma.intan@gmail.com[6], aselamat@utm.my[7]

Abstract. The traditional type of puppet-shadow play, Wayang Kulit, is an integral component of Indonesian culture. The Pandawa Lima, protagonists in this artistic medium, have great importance not just in narrative but also in embodying moral and ethical principles. The automated identification of these characters can optimize a range of applications, such as instructional resources, digital preservation, and interactive displays. This research intends to maximize the advantages of PCA and KNN by utilizing their respective strengths: PCA's capacity to decrease data dimensionality and KNN's efficacy in classification tasks. An expected outcome of this combination is an enhancement in recognition accuracy without compromising computational efficiency. The classification matrix indicates that the model achieved a 78% accuracy rate. Class-specific accuracy, recall, and F1-scores are as follows: arjuna achieves a precision of 0.85, recall of 0.91, and F1 Score of 0.87. Macro averages for precision, recall, and F1 are 0.77, 0.76, and 0.74, respectively. Weighted averages for these metrics are 0.80, 0.78, and 0.77, respectively. The model exhibits strong performances on Arjuna, Sadewa, and Yudistira, but encounters difficulties with Bima and Nakula.

Keywords: Automated recognition; Principal Component Analysis (PCA); K-Nearest Neighbor (KNN); Classification.

1. Introduction

As one of Indonesia's rich cultural heritages, wayang kulit holds a significant role in Indonesian culture. Wayang kulit is not merely entertainment but also a means of moral and ethical education. This traditional Indonesian performing art is rich with philosophical stories. The most famous characters from the Mahabharata wayang tales are the Pandawa Lima, consisting of Yudhishthira, Bima, Arjuna, Nakula, and Sadewa. These five characters are central to the epic Mahabharata narrative. However, the automatic recognition of these characters using technology has not been extensively explored. Automatic recognition of wayang kulit characters not only aids in cultural preservation but also supports education and tourism. This study aims to address this need using the PCA method.

Wayang Kulit is a traditional Javanese shadow puppet theater renowned for its intricate puppetry and rich cultural heritage. Among the characters in Wayang Kulit, the Pandawa Lima, consisting of Yudhishthira, Bhima, Arjuna, Nakula, and Sahadeva, hold significant cultural and historical value. Identifying these characters accurately is essential for preserving and promoting Wayang Kulit. However, manual identification can be challenging due to the detailed and artistic nature of the puppets. Leveraging machine learning techniques, specifically Principal Component Analysis (PCA) and K-Nearest Neighbor (KNN), can automate and enhance the accuracy of this identification process.

Several studies have applied machine learning to image recognition with promising results. For instance, [1], [2] demonstrated the use of PCA in face recognition, highlighting its effectiveness in reducing dimensionality while preserving critical information. Similarly [3] applied KNN in character recognition, proving its robustness in classification tasks [2]. Combining PCA and KNN, [4], [5] showed improved accuracy in recognizing handwritten digits, suggesting the potential of this combination for other forms of image recognition.

In the context of cultural heritage, [6]–[9] utilized convolutional neural networks (CNNs) for the recognition of ancient Indian scripts, indicating the applicability of advanced machine learning techniques in preserving cultural artifacts. Moreover, [10], [11]explored the use of machine learning in identifying traditional dance poses, which parallels the identification of Wayang Kulit characters.

Numerous studies have demonstrated the potential of machine learning in image recognition. [12] pioneered the use of PCA in face recognition, establishing a foundation for subsequent research in this area. [13] highlighted the efficacy of KNN in pattern recognition, which has been widely adopted in various classification problems.

Specific studies that focus on Wayang Kulit character identification include [14] compares the effectiveness of different color channels (Red, Green, Blue) and median filtering in classifying Wayang images using CNN. [15] applies the Backpropagation algorithm in Artificial Neural Networks for recognizing Wayang Kulit patterns. [16] uses the Multi-Layer Perceptron (MLP) method with Gray-Level Co-Occurrence Matrix (GLCM) feature extraction to classify Wayang images. [17] using the k-Nearest Neighbor (k-NN) algorithm combined with GLCM for Wayang image classification. [18] employs transfer learning with CNN models (MobileNetV2 and VGG16) for Wayang image classification. [19] analyzes the accuracy of Wayang image classification using CNN. [20] classifies Pandavas figures in shadow puppetry using a Convolutional Neural Network (CNN), with 430 images across four classes. [21] examines the impact of Gaussian filtering and preprocessing on the classification accuracy of Punakawan puppet images using CNN (VGG16). [22] explores the classification of Wayang images using Support Vector Machine (SVM) and Gray-Level Co-Occurrence Matrix (GLCM) for feature extraction. [23] applies CNN to classify Gagrak (styles) of Wayang Kulit, focusing on Cirebon, Solo, Yogyakarta, and Jawatimuran styles. [24] evaluates the accuracy of Support Vector Machine (SVM) classifiers using different kernel functions—linear, quadratic, and cubic—for classifying Indonesian Wayang images. [25] uses the LeNet architecture to classify Wayang images. [26] presents an image classification system for Wayang Kulit characters using the Extreme Learning Machine (ELM) algorithm and morphological feature extraction. [27] employs Mask R-CNN for recognizing and classifying Indonesian Shadow Puppets. [28] focuses on using Convolutional Neural Networks (CNN) to classify Wanda Janaka, a character from the Ramayana. [29] employs deep learning techniques, specifically AlexNet and VGG-16, for classifying Indonesian shadow puppets. [30] uses CNN on Raspberry Pi 4 to classify Pandawa figures from the Mahabharata story. [31] compares deep learning architectures using transfer learning and fine-tuning for classifying Wayang characters from a small dataset.

Most of the existing studies use CNN, so this study tries to integrate PCA and KNN to classify wayang characters. The main challenge discussed in this study is the accurate identification of Wayang Kulit characters, especially the Pandawa Lima, because the images are almost similar to each other.

The objectives of this research are:

1. To develop an automated system for identifying Pandawa Lima characters in Wayang Kulit using PCA and K-Nearest Neighbor algorithms.
2. To evaluate the performance of the proposed system using a dataset sourced from Kaggle, focusing on accuracy, precision, recall, and F1-score.
3. Comparing the effectiveness of the combined PCA-KNN approach with other machine learning methods previously applied in cultural heritage studies.
4. To provide a scalable framework that can be adapted for the recognition of other Wayang Kulit characters and similar cultural artifacts.

2. Research Method

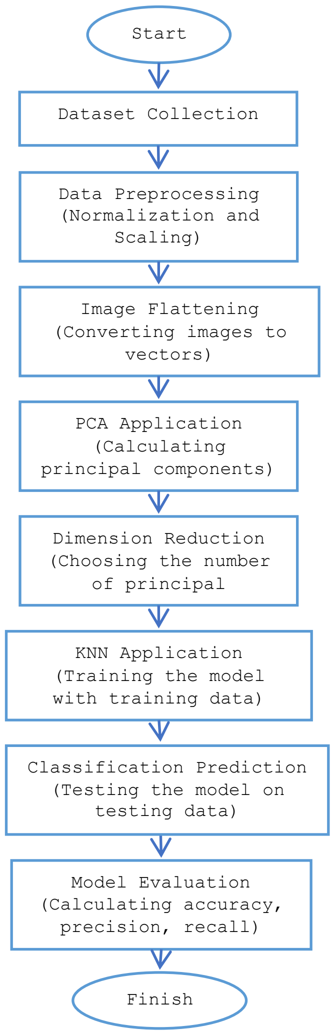

Figure 1. The PCA and KNN

PCA (Principal Component Analysis) is an effective statistical analysis method for pattern recognition. However, its application in the automatic recognition of wayang kulit characters has not been widely explored. PCA is highly effective in reducing data dimensions without losing important information [11][12]. In the context of wayang kulit character recognition, wayang images have high dimensions, which can slow down and make the recognition process inefficient. PCA helps reduce these data dimensions, thereby speeding up the computational process without sacrificing accuracy [13][14].

PCA extracts the main features from images that are most significant for distinguishing one wayang character from another. This allows the recognition system to focus on the truly important aspects of the images, such as facial shapes, clothing, and other distinctive attributes of each Pandawa Lima character. By reducing dimensions and extracting relevant features [15], PCA enhances the accuracy of the recognition system. That is because the features extracted by PCA are the main components that describe the variations in the data, thus making it easier for classification algorithms to recognize and differentiate the wayang characters.

The use of PCA also reduces the number of variables that the recognition algorithm needs to process. This not only saves computational time but also reduces memory requirements, making the system more efficient and capable of running well even on lower-specification devices. PCA provides insights into the data structure by identifying the main patterns and variability in wayang images.

This research aims to classify Puppet images using the Principal Component Analysis (PCA) and K-Nearest Neighbor (KNN) algorithms. The system development approach used is by adapting the waterfall method which includes: requirements analysis, system design, implementation and testing. At the needs analysis stage, Puppet images are collected and analyzed to determine relevant features. Next, at the system design stage, the classification system architecture is designed by considering PCA components for dimension reduction and KNN for classification. The following is the system workflow for classifying puppet images:
The PCA algorithm consists of two main stages: training and testing.

2.1. Training

a. Feature Extraction: Wayang images are analyzed based on color values in RGB and HSV formats and image area size to obtain the main characteristics of each image.
b. Transformation to Principal Components: Principal Component Analysis (PCA) transforms the extracted feature data into principal components. PCA will look for patterns in the data and convert them into new, more compact components.
c. Dimensionality Reduction: The resulting principal components, namely PC1 and PC2, are selected to represent most of the variations in the data. This helps in visualization and improves the efficiency of the classification process.

2.2. Testing

a. Feature Extraction of Test Images: Wayang images to be tested are processed using the same method as in the training stage, namely RGB, HSV, and area size feature extraction.
b. Projection to Principal Component Space: Features from the test image are transformed into the principal component space previously created in the training stage, using the same PCA parameters.
c. Classification with KNN: The principal components of the test image are compared with the training data using the K-Nearest Neighbors (KNN) algorithm. KNN determines the class of a test image based on the proximity to the nearest sample in the principal component space.
d. Result Evaluation: The system's performance is assessed by comparing the classification results with the correct labels, using metrics such as accuracy, precision, recall, and F1 score to determine how well the model performs.

3. Results and Discussion

Application of Principal Component Analysis (PCA) and K-Nearest Neighbors (KNN) for picture categorization with dataset folder Constructing dataset from the Dataset Collection directory. Normalization and scaling of data using preprocessing techniques. Image Flattening is the process of converting images into vector spaces. Principal Component Analysis (PCA) Application (calculating principal components), Dimensionality Reduction (determining the number of principal components), K-

Nearest Neighbors (KNN) Application (training the model using training data), Classification Prediction (applying the model to testing data), Model Evaluation (calculating accuracy, precision, recall).

3.1. Wayang Dataset

This research obtained the Wayang dataset mainly from Kaggle and Google Image Search. The dataset has various backgrounds and lighting conditions. The dataset consists of 3,000 wayang images divided into 5 classes. The classes are wayang characters, namely Arjuna (Janaka), Bima (Werkudara), Nakula, Sadewa, and Yudhistira (Puntadewa). They are the most important characters in the Mahabharata stories [32].

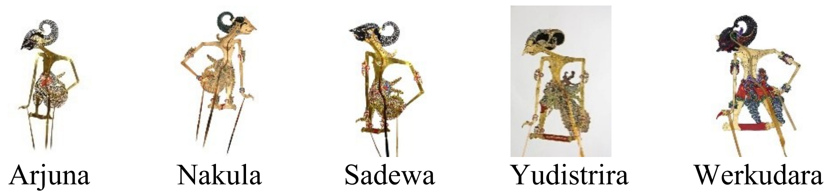

Figure 2. The Pandawas Lima

3.2. Collation of Datasets

The objective of Dataset Collection is to import and manipulate digital photographs from the dataset directory to be used in machine learning models. Initially, the code imports crucial libraries such os for managing files, NumPy for manipulating arrays, and PIL.Image for processing images. The load_images_from_folder function manipulates images by accessing each image placed in the class subdirectory, transforming it into grayscale format, adjusting its size to the given size, and converting it into a one-dimensional array. Next, the photos and their corresponding class labels are saved in a numpy array. After defining the function, the code establishes the path to the dataset and imports the picture data and labels using the function. This prepares the data for use in the training of a classification model.

```
Class 'arjuna' has 46 image
Class 'bima' has 39 image
Class 'yudistira' has 15 image
Class 'nakula' has 38 image
Class 'sadewa' has 19 image
Total gambar di data validation: 157
Class 'bima' has 100 image
Class 'nakula' has 100 image
Class 'arjuna' has 250 image
Class 'sadewa' has 100 image
Class 'yudistira' has 100 image
Total images in testing data: 650
```

Figure 3. The Dataset

3.3. Data Preprocessing - Normalization and Scaling

After converting the image into an array, it is necessary to normalize it by dividing each pixel value by 255. This may be achieved using the scikit-learn module to appropriately normalize or scale the data. This step is crucial in data preparation prior to training a machine learning model. The first step is importing StandardScaler from sklearn. preprocessing, a class that standardizes data features by converting them into a distribution with a mean of 0 and a common variance of 1. After instantiating a StandardScaler object and assigning it to the scaler variable, the code proceeds to apply the fit_transform() method to the data array X. By applying a transformation to each data feature, this method computes the mean and standard deviation of the data X, resulting in a mean of 0 and a standard deviation of 1. The normalized data obtained is saved in the X_scaled variable, which is designated for usage in model training that may necessitate data in a uniform scale to attain the best possible performance.

3.4. Implementation of PCA and KNN

Following the process of dimensionality reduction using Principal Component Analysis (PCA), we can proceed to train the K-Nearest Neighbours (KNN) model by integrating two machine learning approaches, namely PCA and K-1. Principal Component Analysis (PCA) is employed to decrease the dimensionality of the dataset and to decrease the number of features while therefore preserving pertinent information. main Component Analysis (PCA) limits the dataset to 50 main components. Subsequently, the modified dataset is partitioned into training data and test data via the train_test_split function, allocating 70% of the data for training and the remaining 30% for testing. Moreover, KNN, a supervised learning technique, is employed for undertaking classification tasks. In this instance, the KNN model is trained with the parameter k = 3, indicating that the model would seek the three closest neighbors to infer the class of a new sample. Integrating Principal Component Analysis (PCA) with K-Nearest Neighbors (KNN) enables the model to operate with more efficiency in a space with fewer dimensions, while yet preserving strong data categorization skills.

3.5. Prediction and Evaluation

Approximation of predictions using experimental data and assessment of the effectiveness of the K-Nearest Neighbours (KNN) model following its training with training data. Based on the test data (X_test), the model generates class predictions (y_pred). After comparing these predictions to the original labels of the test data (y_test), several evaluation metrics are computed. The accuracy_score function is used to compute accuracy, quantifying the proportion of accurate predictions among the entire test data. Furthermore, the precision_score and recall_score functions are used to compute precision and recall. The average='weighted' argument calculates a weighted average by considering the number of samples in each class. Precision quantifies the level of accuracy with which the model predicts the proper class, whereas recall quantifies the capacity of the model to identify all occurrences of that class. Lastly, a classification report is presented to summarize other metrics like the F1-score for each class and provide a comprehensive overview of the model's performance in the classification assignment.

```
Akurasi: 0.78
Presisi: 0.80
Recall: 0.78
              precision    recall  f1-score   support

      arjuna       0.85      0.91      0.87        85
        bima       0.88      0.39      0.54        36
      nakula       0.41      0.50      0.45        26
      sadewa       0.87      1.00      0.93        20
   yudistira       0.85      1.00      0.92        28

    accuracy                           0.78       195
   macro avg       0.77      0.76      0.74       195
weighted avg       0.80      0.78      0.77       195
```

Figure 4. The Accuracy

The classification report is a tabular representation of the performance evaluation findings of the classification model. It presents several metrics that demonstrate the overall performance of the model as well as its performance within each individual class. First, Accuracy quantifies the proportion of accurate forecasts among all the predictions produced. In this instance, the model attained an accuracy of 0.78, indicating that it made correct predictions in 78% of the total samples examined. For each class, including arjuna, bima, nakula, sadewa, and yudistira, the report presents the precision metric, which indicates the level of accuracy of the model in predicting for that specific class. Specifically, the precision for the class "arjuna" is 0.85, meaning that 85% of all predictions made as "arjuna" were accurate. The recall metric measures the model's ability to correctly identify all instances of a specific class. In the case of the class "arjuna", the recall is 0.91, indicating that 91% of all "arjuna" samples were correctly identified. The F1-Score is calculated as the harmonic mean of precision and recall, providing a more equitable representation of performance in cases when imbalances exist between the two. The F1-score of "arjuna" was measured to be 0.87. Support is the quantification of the number of samples for each class in the test data. In the case of the class "arjuna", the support is 85, which means that there are exactly 85 observations in that particular class. Furthermore, the report also presents the macro statistical average and weighted average. The macro average is a statistical measure that calculates the mean of accuracy, recall, and F1-score metrics over all classes, disregarding the amount of samples in each class. The mean values for precision, recall, and F1-score at a macro level are 0.77, 0.76, and 0.74, respectively. Conversely, the weighted average considers the amount of samples in each class, providing a more accurate representation of overall performance, evidenced by precision values of 0.80, recall of 0.78, and F1-score of 0.77.

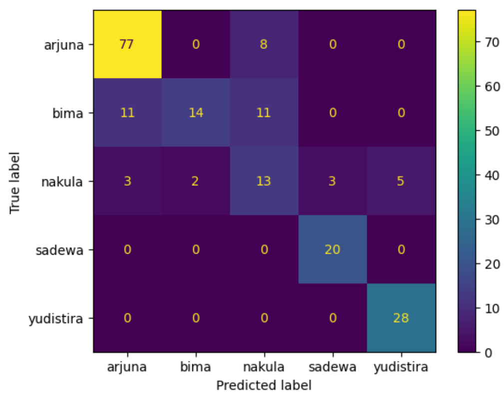

Figure 5. Confusion matrix

The displayed graphics are a confusion matrix, which offers a comprehensive representation of the classification model's performance by illustrating its predictions of the class for each case in the dataset. The vertical axis corresponds to the initial label (actual class) of the dataset, and the horizontal axis corresponds to the predicted label generated by the model. Every individual cell within the matrix displays the count of projected cases for a specifically designated class. In the given cell (0,0), there are precisely 77 instances of the arjuna class that are accurately identified as arjuna. The primary diagonal stretching from the top left to the bottom right displays the count of accurately classified instances. Specifically, there are 14 instances of the Bima class, 13 instances of the Nakula class, 20 instances of the Sadewa class, and 28 instances of the Yudistira class, all of which are correctly classified.

However, the cells that are not diagonal display the count of incidents that have been incorrectly classified. For instance, there are 11 cases of the Bima class that have been incorrectly categorized as arjuna, and 8 cases of the arjuna class that have been accurately categorized as Nakula. Based on this matrix, it can be inferred that the model exhibits a high level of accuracy in some categories such as Arjuna, Sadewa, and Yudistira, as the majority of cases belonging to these categories are accurately identified. However, the model exhibits more pronounced inaccuracies when applied to classes such as Bima and Nakula, resulting in numerous misclassifications of these classes as other classes. Furthermore, beyond the confusion matrix, complementary measures such as accuracy, recall, and F1-score can be employed to offer a more comprehensive assessment of the model's performance, particularly when dealing with an imbalanced dataset.

This study applied PCA and KNN with 78% accuracy, indicating that these methods can reduce the dimensionality of data while retaining important information for classification. Compared with CNN-based methods, such as dropout on CNN (accuracy 93%) [18] and transfer learning with MobileNetV2 and VGG16 (accuracy >94%) [20], PCA and KNN excel in computational efficiency but still face challenges in distinguishing classes with similar visual features. Combining PCA and CNN can be a promising approach to improving performance. PCA is used for dimensionality reduction before the features are fed into CNN, thus accelerating training without losing important information. In addition, optimization of the number of principal components (PCs), exploration of LDA or deep learning-based feature selection, and improvement of KNN with weighted KNN or optimal distance selection can improve the accuracy of the PCA-based model. With this strategy, the wayang image recognition system can be more accurate and efficient and contribute to preserving digital culture in Indonesia.

4. Conclusion

Classification matrix showing the evaluation of the performance of the classification model. The accuracy of the model is 0.78, which means 78% of the predictions are correct. The report includes precision, recall, and F1-score metrics for each class (Arjuna, Bima, Nakula, Sadewa, Yudistira). For example, the arjuna class has a precision of 0.85 and a recall of 0.91, with an F1-score of 0.87. The macro average shows a precision of 0.77, a recall of 0.76, and an F1-score of 0.74, while the weighted average is 0.80 for precision, 0.78 for recall, and 0.77 for F1-score. The model performs well on the Arjuna, Sadewa, and Yudistira classes, but is less effective on Bima and Nakula, indicating the need for improvement in these areas.

The confusion matrix illustrates the classification model's performance by comparing actual class labels (vertical axis) with predicted labels (horizontal axis). It shows counts of correct predictions on the diagonal (e.g., 77 for Arjuna, 14 for Bima), and misclassifications off the diagonal (e.g., 11 Bima cases mislabeled as Arjuna). The model performs well with Arjuna, Sadewa, and Yudistira but struggles with Bima and Nakula, leading to more misclassifications. For a thorough evaluation, especially with imbalanced datasets, additional metrics like accuracy, recall, and F1-score should be used.

For further research, data augmentation is recommended to handle class imbalance, combining PCA and CNN to improve efficiency and accuracy and transfer learning with more complex models such as EfficientNet or ResNet. With these improvements, the system can be more accurate in recognizing wayang and contribute to preserving digital culture.

5. References

[1] L. C. Paul and A. Al Sumam, "Face recognition using principal component analysis method," *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 1, no. 9, pp. 135–139, 2012.

[2] R. Kaur and E. Himanshi, "Face recognition using principal component analysis," in *2015 IEEE international advance computing conference (IACC)*, 2015, pp. 585–589, doi: 10.1109/IADCC.2015.7154774.

[3] V. Ong and D. Suhartono, "Using k-nearest neighbor in optical character recognition," *ComTech Comput. Math. Eng. Appl.*, vol. 7, no. 1, pp. 53–65, 2016, doi: 10.21512/comtech.v7i1.2223.

[4] M. S. A. Malik, N. Kousar, T. Abdullah, M. Ahmed, F. Rasheed, and M. Awais, "Pakistan sign language detection using PCA and KNN," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 4, 2018.

[5] M. Baheti, "KNN and PCA Approach for Gujarati Script," *Nanotechnol. Perceptions*, pp. 388–394, 2024, doi: 10.62441/nano-ntp.v20iS6.27.

[6] M. Agarwal, S. Indu, and N. Jayanthi, "An Approach to the Classification of Ancient Indian Scripts Using the CNN Model," in *International Conference on Women Researchers in Electronics and Computing*, 2023, pp. 367–377, doi: 10.1007/978-981-99-7077-3_36.

[7] R. Krithiga, S. R. Varsini, R. G. Joshua, and C. U. O. Kumar, "Ancient character recognition: a comprehensive review," *IEEE Access*, 2023, doi: 10.1109/ACCESS.2023.3341352.

[8] L. Giridhar, A. Dharani, and V. Guruviah, "A novel approach to ocr using image recognition

based classification for ancient tamil inscriptions in temples," *arXiv Prepr. arXiv1907.04917*, 2019, doi: 10.48550/arXiv.1907.04917 Focus to learn more.

[9] D. R. Devi, M. D. Kanna, and S. Rajendran, "Ancient tamil digits recognition using convolutional neural network," in *AIP Conference Proceedings*, 2024, vol. 3180, no. 1, doi: 10.1063/5.0224624.

[10] S. Gupta and S. Singh, "Indian dance classification using machine learning techniques: A survey," *Entertain. Comput.*, p. 100639, 2024, doi: 10.1016/j.entcom.2024.100639.

[11] J. Shen and L. Chen, "Application of human posture recognition and classification in performing arts education," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3451172.

[12] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*, 1991, pp. 586–587, doi: 10.1109/CVPR.1991.139758.

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2017.

[14] 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*, 2023, vol. 2706, no. 1, doi: 10.1063/5.0120384.

[15] K. A. Nugraha, A. J. Santoso, and T. Suselo, "Algoritma backpropagation pada jaringan saraf tiruan untuk pengenalan pola wayang kulit," in *Seminar Nasional Informatika (SEMNASIF)*, 2015, vol. 1, no. 4.

[16] M. H. Santoso, D. A. Larasati, and M. Muhathir, "Wayang Image Classification Using MLP Method and GLCM Feature Extraction," *J. Comput. Sci. Inf. Technol. Telecommun. Eng.*, vol. 1, no. 2, pp. 111–119, 2020, doi: 10.30596/jcositte.v1i2.5131.

[17] B. Sandy, J. K. Siahaan, and P. Permana, "Klasifikasi Citra Wayang dengan Menggunakan Metode K-NN dan GLCM," in *Seminar Nasional Teknologi Informatika (Semantika)*, 2021, pp. 71–77.

[18] M. Banjaransari and A. Prahara, "Image Classification of Wayang Using Transfer Learning and Fine-Tuning of CNN Models," *Bul. Ilm. Sarj. Tek. Elektro*, vol. 5, no. 4, pp. 632–641, 2023, doi: 10.12928/biste.v5i4.9977.

[19] M. R. A. Yudianto, K. Kusrini, and H. Al Fatta, "Analisis Pengaruh Tingkat Akurasi Klasifikasi Citra Wayang dengan Algoritma Convolitional Neural Network," *J. Teknol. Inf.*, vol. 4, no. 2, pp. 182–191, 2020, doi: 10.36294/jurti.v4i2.1319.

[20] W. Supriyanti and D. A. Anggoro, "Classification of pandavas figure in shadow puppet images using convolutional neural networks," *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 7, no. 1, 2021, doi: 10.23917/khif.v7i1.12484.

[21] K. Kusrini, 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," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 4, p. 3752, 2022, doi: 10.11591/ijece.v12i4.pp3752-3761.

[22] M. Muhathir, M. H. Santoso, and D. A. Larasati, "Wayang Image Classification Using SVM Method and GLCM Feature Extraction," *J. Informatics Telecommun. Eng.*, vol. 4, no. 2, pp. 373–382, 2021, doi: 10.31289/jite.v4i2.4524.

[23] A. S. S. Pratama, A. P. Wibawa, and A. N. Handayani, "Convolutional Neural Network (Cnn) Untuk Menentukan Gagrak Wayang KULIT," *J. Mnemon.*, vol. 5, no. 2, pp. 98–102, 2022, doi: 10.36040/mnemonic.v5i2.4671.

[24] Muhathir, "Measuring the accuracy of SVM with varying Kernel function for classification of Indonesian Wayang on Images," in *2020 International Conference on Decision Aid Sciences and Application (DASA)*, 2020, pp. 1190–1196, doi: 10.1109/DASA51403.2020.9317197.

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

[26] F. Fatmayati, M. Nugraheni, R. Nuraini, and F. Rossi, “Classification of Character Types of Wayang Kulit Using Extreme Learning Machine Algorithm,” *Build. Informatics, Technol. Sci. Vol.*, vol. 5, pp. 317–326, 2023, doi: 10.47065/bits.v5i1.3568.

[27] 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, doi: 10.1088/1742-6596/1783/1/012032.

[28] B. H. Suharto and M. Hardiyanti, “Identifikasi Wanda Janaka berbasis Deep Learning dengan Metode Convolutional Neural Network,” *Comput. Sci. Res. Its Dev. J.*, vol. 15, no. 3, 2023.

[29] I. B. K. Sudiatmika, “Indonesian traditional shadow puppet image classification: A deep learning approach,” in *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2018, pp. 130–135, doi: 10.1109/ICITEED.2018.8534776.

[30] K. Wisnudhanti and F. Candra, “Image classification of pandawa figures using convolutional neural network on raspberry pi 4,” in *Journal of Physics: Conference Series*, 2020, vol. 1655, no. 1, p. 12103, doi: 10.1088/1742-6596/1655/1/012103.

[31] 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, doi: 10.14421/ijid.2020.09207.

[32] A. Kaelola, *Mengenal Tokoh Wayang Mahabharata*. Media Pressindo, 2010.