JISS Journal of Indonesian Social Science

Vol. 6, No. 5, May 2025
E-ISSN:2723 – 6692
P-ISSN:2723– 6595
http://jiss.publikasiindonesia.id/

Classification of Wayang Kulit Using Canny Feature Extraction and Convolutional Neural Network Algorithm

Asep Rudi, Riza Ibnu Adam
Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang, Indonesia
Email: 1910631170165@student.unsika.ac.id

KEYWORDS

Ingenious edge detection, DenseNet-121, classification, leather stuffing

ABSTRACT

Wayang kulit is a part of Indonesian culture known to the Javanese people, but the younger generation often has difficulty recognizing the wayang characters they are looking for online because of inaccurate search results. One popular story is the Mahabharata, with the characters of the Five Pandavas: Puntadewa (Yudistira), Bima, Arjuna, Nakula, and Sadeva. Because puppet characters have similar shapes, curves, clothing, and colors, it is often difficult to distinguish and remember. This shows the need for technology to help recognize puppet characters more easily. This research aims to solve this problem by utilizing Deep Learning techniques in Computer Vision to classify puppet images. Canny's feature extraction technique and DenseNet-121 architecture are used to detect patterns in the puppet image and classify them into appropriate categories. The dataset used consisted of 1028 images divided into four categories: Arjuna, Bima, Nakula & Sadewa, and Puntadewa. The framework implemented is CRISP-DM, with the implementation using the Python 3.11 programming language, TensorFlow 2.14, and the Google Colab tool. The results of the model evaluation through the confusion matrix showed 93% accuracy, 93% precision, 93% recall, and 92% f1 score. With these results, it is hoped that technology can facilitate and increase accuracy in recognizing the character of puppet puppets.

Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)

INTRODUCTION

Wayang kulit is one part of Indonesian culture that has been widely known by the Javanese people (Muhathir et al., 2021a). Wayang kulit has been recognized by UNESCO as "*Masterpiece of Oral and Intangible Heritage of Humanity*" on November 7, 2003. Citizens with integrity are expected to preserve it and always pass it on to future generations (Muhathir et al., 2021b). The wayang kulit culture that was once known by many figures is now only a small part of it is still known by the public. This is due to three main factors that are weak, namely the artists who work, the groups that maintain the puppet culture, and the lack of support from the government (Mustafid et al., 2020). People look for puppet puppet figures on the internet that are produced, namely the images and names of characters that do not match, making it difficult for people to know them,

this culture over time will become extinct, and many of the next generations will lose knowledge about puppet characters (Fatmayati et al., 2023a).

One of the puppet characters that are familiar to the people in Indonesia is the story of Mahabharata (Bastian, 2023). The Mahabharata teaches many values about human character, including honesty, virtue, and how to deal with evil. This story is not just a legend, but an exploration of the depth and complexity of the human soul. The Mahabharata tells the journey of the Bharata family involved in a major conflict, the Bharatayuda war, which occurred between fellow family members (Sutama, 2023). Among the many characters in the Mahabharata story, there is a very famous group of protagonists, namely the Pandavas Lima. The Pandavas are a group of famous brothers in the Mahabharata story, consisting of five members with unique names and characteristics, namely Puntadewa or Yudistira, Bima, Arjuna, Nakula, and Sadewa (Raharjo & Ningrum, 2024a). The embodiment of puppet puppet characters in the world of puppetry is depicted with similar shapes and curves, as well as the clothes and colors used by each puppet character. Because of this resemblance, puppet characters are often difficult to distinguish and remember (Prabowo et al., 2021a). This indicates the need for technology that uses *Deep Learning* deep *Computer Vision* To facilitate the recognition of puppet figures, by detecting patterns in the image and classifying the puppet image. The image of the puppet used is the character of the five pandavas in the story of Mahabaratha.

Edge detection research for pattern recognition and classification in imagery has been conducted by Vipul et al., (2021). This study introduces patterns in Indian Sign Language (ISL) letter images and classifies them into alphabets, the methods used are *canny* for edge detection and utilizing the *Convolutional Neural Network* (CNN) to classify *Dataset* Indian Sign Language letters and *Dataset* alphabet. The results showed that the accuracy of *canny edge detection* 98% and CNN architecture achieves accuracy *training* by 95.9% and accuracy *validation* by 98.5%. However, the *canny* has a weakness, namely a long computing time, which is 4 hours in the training process. Previous research was conducted by Bello et al., (2024a) Exploring the impact *machine learning* Traditional and Approach *deep learning* which is refined on prediction accuracy. Type *transfer learning* which are enhanced such as *EfficientNetB0, ResNet34, VGG16, Inception_v3,* and *DenseNet121* show superior performance. The results of the study on the classification of skin cancer using *DenseNet*-121 gets 87% accuracy.

Based on the discussion, it is known that the application of *the canny edge detection method* has good performance for detecting the edges of a pattern on *images* and the *Convolutional Neural Network* method with *DenseNet-121* architecture as a method for classifying *images.* This study uses *the python* programming language and *the TensorFlow library* which has good performance to classify based on image data. Thus, the research to be carried out is entitled "Classification of Wayang Kulit Using *Canny Feature Extraction* and *Convolutional Neural Network Algorithm*".

The purpose of this study is to apply the Canny Edge Detection Feature method and the CNN (Convolutional Neural Network) algorithm with the DenseNet-121 architecture in classifying the images of Pandawa Lima puppet characters, namely Arjuna, Bima, Nakula & Sadewa, and Puntadewa, as well as to determine the accuracy of the classification of puppet images using this method. The benefits of this research are divided into two, namely theoretical and practical

benefits. Theoretically, this study provides enlightenment and can be a guide for further research related to the working of the Canny Feature Extraction method and the Convolutional Neural Network algorithm with the DenseNet-121 architecture in classifying four puppet figures, especially the Pandawa characters of the Five classes Arjuna, Bima, Nakula & Sadewa, and Puntadewa. Meanwhile, the practical benefit of this study is to determine the application of Canny Feature Extraction in detecting the edges of puppet images and to find out how high the accuracy level of the Convolutional Neural Network algorithm with DenseNet-121 architecture is in detecting puppet objects in a puppet image with the classes used, namely Arjuna, Bima, Nakula & Sadewa, and Puntadewa.

RESEARCH METHODS

The object of this research is puppetry, with the dataset used consisting of puppet images of four prediction classes, namely Five Pandawa figures: Arjuna, Bima, Nakula & Sadewa, and Puntadewa. This dataset was obtained from the Kaggle website under the name "Indonesian Wayang (Traditional Puppet Show)" in JPG/JPEG format, which contains images of characters from the Mahabharata story. This study aims to classify puppet images using the Canny Feature Extraction method and the Convolutional Neural Network (CNN) algorithm with the DenseNet-121 architecture. The methodology of this study adopts CRISP-DM which consists of several stages, namely Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. At the Business Understanding stage, the problem found was the difficulty of the community in recognizing puppet puppet characters due to the inconsistency of images and names of figures circulating on the internet. In Data Understanding, data collection and dataset exploration were carried out to find out the amount of data in each class of puppet characters. Data Preparation includes steps such as data loading, cleaning, normalization, one-hot encoding, data augmentation, and data sharing for training, validation, and testing. In the Modeling stage, the DenseNet-121 model is used for image classification, and in the Evaluation stage, tests are carried out with new data to measure the performance of the model. The results of the research will be compiled in reports and journal articles to be disseminated.

RESULTS AND DISCUSSION

Research Results

The results of this study describe the application of *Canny feature extraction* in the DenseNet121 architecture for the classification of puppet images, which is carried out using the Python programming language and *Google Colab tools*. This study uses the CRISP-DM framework which consists of business *understanding, data understanding, data preparation, modeling, evaluation,* and *deployment.*

Business Understanding

At this stage, a business understanding is carried out by explaining the existing problems, which then helps in determining the research objectives. The problem faced in this study is that people have difficulty recognizing puppet puppet characters because search results on the internet often display inappropriate images and character names. This has the potential to cause the puppet

culture to decline, and many of the next generations do not know puppet characters (Fatmayati et al., 2023b). The puppet characters in the world of puppetry are depicted with almost similar shapes, curves, clothes, and colors, making them difficult to distinguish and remember (Prabowo et al., 2021b). Therefore, the purpose of this study is to make it easier to recognize puppet characters through classification. The puppet figures used in this study as a dataset class include Arjuna, Bima, Nakula & Sadewa, and Puntadewa.

Data *Understanding*

Data understanding is an important step in analyzing and understanding the data used in research. The dataset used comes from the Kaggle website under the name "Indonesian Wayang (Traditional Puppet Show)" and consists of 22 puppet characters from the Mahabharata story, including Arjuna, Bima, Nakula, Sadewa, and Puntadewa, with a total of 6,576 images in .jpg format. In this study, the focus of classification is limited to five Pandawa figures, namely Arjuna, Bima, Nakula, Sadewa, and Puntadewa, which totals 1,028 images with the following details: 400 images for Arjuna, 336 images for Bima, 192 images for Nakula_Sadewa, and 272 images for Puntadewa (Raharjo & Ningrum, 2024b). This dataset is used for analysis using Python and Google Colab.

The next stage is to perform Exploratory Data Analysis (EDA), which aims to understand the characteristics of the data and prepare the data for further modeling. The EDA process includes creating an image histogram to analyze the brightness distribution, calculate the sharpness of the image, and detect noise. In addition, random image visualization is carried out to compare the variation of images in the dataset and see the pattern of the dataset. The EDA process also analyzes the distribution of image sizes, with the result that most images are 700 pixels wide and 900 pixels high, indicating medium to high resolution quality. Other results include a color distribution dominated by red, which is important for the visual representation of the puppet (Isa et al., 2023).

In addition, image quality is analyzed to assess sharpness, blurriness, and noise. The results of the analysis showed that most of the images had sharp quality, although there were variations in sharpness levels. The EDA process also includes checking for damaged images and checking the lighting and sharpness of the images. It was found that there were 106 images with abnormal lighting and 15 blurry images, which were then considered in further data processing. Using Canny's edge detection method, the images are processed to identify the edges of objects, which helps to improve accuracy in the classification process. The Canny method was chosen because it has been proven to provide sharper and more accurate results in edge detection than other methods (Khoirani & Ariansyah, 2024)..

Data *Preparation*

Data *preparation* is the process of preparing data by adjusting the dataset to suit the needs at the modeling stage. Based on the results of the data understanding, it was found that in the dataset used there were 106 data with abnormal lighting, 15 blurry image data based on sharpness, and 15 blurry image data. Therefore, the data cleaning and edge detection process is carried out using *the Canny* method.

On the data process *cleaning,* The first stage creates a DataFrame to display the dataset used with the *Image Name, Folder, Width, Height* and *Format.* A total of 1208 rows. DataFrame results image details are displayed on Figure .

```
Detail Gambar:
                Image Name     Folder  Width  Height Format
0            arjuna010.jpg     arjuna    225     225   JPEG
1            arjuna006.jpg     arjuna    311     554   JPEG
2            arjuna002.jpg     arjuna    183     275   JPEG
3            arjuna012.jpg     arjuna    720     887   JPEG
4            arjuna015.jpg     arjuna    412     685   JPEG
...                    ...        ...    ...     ...    ...
1203      puntadewa010.jpg  puntadewa    340     556   JPEG
1204  puntadewa001h (3).jpg  puntadewa    633     800   JPEG
1205      puntadewa006.jpg  puntadewa    185     273   JPEG
1206      puntadewa011.jpg  puntadewa    182     277   JPEG
1207            img012.jpg  puntadewa    720     864   JPEG

[1208 rows x 5 columns]
```

Figure 1. Detailed Images

Furthermore, the total data from each class in the dataset used is displayed in the Figure.

```
Jumlah Gambar per Kelas:
      Nama Kelas  Jumlah Gambar
0         arjuna            399
1           bima            346
2  nakula_sadewa            191
3      puntadewa            272

Total Jumlah Gambar: 1208
```

Figure 2. Total Images in Each Class

The dataset was carried out *load image* using the setting i.e. taking *Image* and *class* from the puppet dataset with a display image size of 224x224 pixels. Then, convert the image to *Array* and implement edge detection using *Canny.* Next, it converts the edge detection results into a format suitable for display (into a color image with three channels. In addition, it handles errors in case of problems loading images. Subplot to display images as well as draw each image that has been processed, give titles with labels, and hide axes. Result *load image* The puppet dataset is displayed on the Figure .

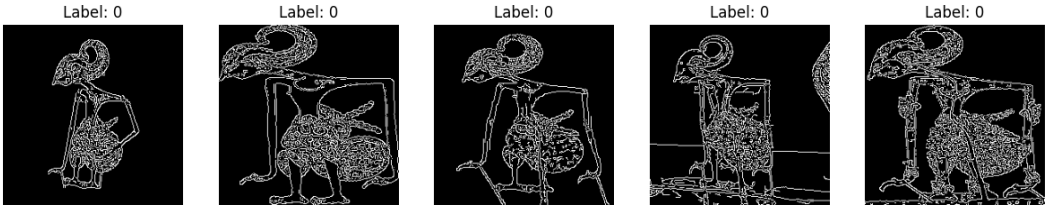

Figure 3. Wayang Kulit Dataset Per Class

The next stage is to check the classes and labels of the image dataset contained in a specific directory. The class used is changed in the form of an index shown in the Figure .

```
Jumlah kelas: 4
Label dictionary: {'arjuna': 0, 'bima': 1, 'nakula_sadewa': 2, 'puntadewa': 3}
Shape of X: (1210, 224, 224, 3)
Shape of y: (1210,)
Unique labels: [0 1 2 3]
Sample labels: [0 0 0 0 0 0 0 0 0 0]
```

Figure 4. Index Dataset Results

After the detection *of Canny*, data normalization was carried out on the dataset. Next, encoding is carried out using one-hot encoding. The next data preparation process is data augmentation with the following settings: *width_shift_range=0.2* to move the image horizontally up to 20% of the width of the original image, *height_shift_range=0.2* to move the image vertically up to 20% of the height of the original image, *zoom_range=0.2* to zoom in on the image by zooming in or out by up to 20%, and *fill_mode='nearest'* to fill in the missing pixels when performing a transformation (for example, when the image is shifted). Using the *'nearest' setting*, the missing pixel will be populated with the value of the nearest pixel. After that, the mean and standard deviation values from the training dataset are calculated.

After the augmentation is carried out, the data is divided into *training* data and temporary data. The data sharing consists of 80% for training data and 20% for temporary data. Of the 20% of the provisional data, 10% is used for data validation and the remaining 10% is for data *testing*.

After the stage *preprocessing* which includes data *cleaning*, edge detection using the *Canny*normalization *one-hot encoding*, and augmentation, as well as data sharing (*Split*), the resulting dataset becomes clean, structured, balanced, and ready to be used in the training of the image classification model. This data has been cleaned, scaled, and enriched to ensure the model can learn effectively. Thus, the dataset is ready to be used for image classification using *Convolutional Neural Network* (CNN). The CNN algorithm is part of the supervised learning method that functions to identify images by training a model using processed image data and setting targets on those image variables (Alfiansyah & Litanianda, 2024a). The CNN model applied is the DenseNet121 architecture. DenseNet121 connects each output layer to the next, utilizing that output as input. This helps improve the flow of information between layers and reduces the number of parameters required (Alfiansyah & Litanianda, 2024b). The DenseNet121 model, which has thousands of parameters, has proven to be very effective in recognizing objects in imagery (Fadillah, 2025). In addition, DenseNet-121 has been shown to have excellent performance on a wide range of image classification tasks (Listyalina et al., 2021). Therefore, DenseNet121 will be used in modeling for the classification of puppet images.

Modeling

At the *modeling* stage, it was carried out to form the DenseNet121 model using *the best model* from DenseNet121. *The library* used is a model to create a hard model class. *The Input Library, Dense, Dropout, GlobalAveragePooling2D* are the various layers used in the model. *The* DenseNet121 library is a DenseNet121 architecture model that has been pre-trained on *the ImageNet dataset. Adam optimizer* for model training.

In the model, the transfer learning technique is selected, in this case, DenseNet121 by importing *the model* and replacing the last layer of the model. The DenseNet121 model uses *weights='imagenet'*, which means it takes advantage of the pre-trained weights on the *ImageNet dataset*. *include_top=False* to ignore the top classification layer of the original DenseNet121 model, so that it can add its own classification layer. *input_shape*=(224, 224, 3) to specify the image *input* shape to be processed by the model (224x224 pixels with 3 color channels). Then, define *the inputs* and *outputs*. Next, add a classification layer consisting of *GlobalAveragePooling2D* to reduce the *output* dimensions of the *base* model to vectors by averaging features across spatial dimensions. *Dense(512, activation='relu')* to add a *Dense* layer with 512 *neurons* and the ReLU activation function. Dropout Layer(0,2) to add a dropout layer to reduce *overfitting* by ignoring 20% of *neurons* during training. The *Dense(6, activation='softmax')* layer as the *output* layer has 6 *neurons* (according to the number of classes) and the softmax activation function for multi-class classification. Then, build the model by defining *predefined inputs* and *outputs* and setting all layers of the *base model* (DenseNet121) so that they cannot be trained (*frozen*) during training, so that only additional layers are trained. This is useful for taking advantage of existing features without changing the *pre-trained weight.*

Models that have been created, done *generate test design* to produce DenseNet121 modes that are ready to be used in classification. Then, it is carried out *compile model* to produce *best model* Using *hyperparameter tuning* that is *learning_rate 0.0001, epoch 100, optimizer Adam, loss categorical_crossentropy* and *metric accuracy.* Once specified *hyperparameter tuning,* then it is carried out *training model.* A summary of the model is displayed Figure .

Model: "functional_1"

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 224, 224, 3)	0
densenet121 (Functional)	(None, 7, 7, 1024)	7,037,504
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 1024)	0
dense_2 (Dense)	(None, 512)	524,800
dropout_1 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 4)	2,052

Total params: 7,564,356 (28.86 MB)
Trainable params: 526,852 (2.01 MB)
Non-trainable params: 7,037,504 (26.85 MB)

Figure 5. Results of the formation of the DenseNet121 model

The DenseNet121 model consists of several layers, starting with an input layer that accepts images with dimensions of 224x224 pixels and 3 color channels. Then, the DenseNet121 layer reduces the image size to 7x7 pixels with 1024 features. Global Average Pooling converts 3D tensors into 1024-sized 1D vectors with no trained parameters. Next, there are two layers of dense, the first with 512 neurons and the second producing 4 output classes. Dropout layers are used to

prevent overfitting. The total parameters of the model reached 7,564,356, with 526,852 trainable parameters and 7,037,504 non-trainable parameters, which are the parameters of the pre-trained DenseNet121 model. The model combines extraction, regularization, and classification features with parameters that have been optimized for image recognition tasks.

Evaluation

In the training stage of the DenseNet121 model with Canny feature extraction, the results of the training history show that the model achieves 93% accuracy, 93% precision, 93% recall, and 92% F1-score. The training was carried out with 100 epochs, batch size 64, and produced accuracy and loss graphs that showed a steady improvement during the training process. The accuracy graph shows a significant increase in training and validation, with a final accuracy value of 0.9946 in training and 0.9245 in validation. The loss graph shows a consistent decrease in the value of losses, reflecting the "good fit" condition, where the values of training loss and validation loss are both low. The new data experiment with Nakula_Sadewa and Arjuna images shows fast prediction times, with high confidence. Evaluation using the confusion matrix revealed that the model successfully classified the data with 93% accuracy, 93% precision, 93% recall, and 92% F1-score. The results of the calculations for precision, recall, and F1-score in each class also showed good performance, with the average scores for precision, recall, and F1-score reaching 93%, 93%, and 92%, respectively.

Deployment

At the *deployment* stage, reports are prepared and journal articles are written using the model that has been developed. The results of the model performance performance in the classification of puppet images using *Canny feature extraction* in the DenseNet-121 architecture that have been evaluated will be made a report of the research results in the form of knowledge information.

In classifying puppets that apply feature extraction *canny* on the CNN algorithm with the DenseNet121 architecture using the framework of the CRISP-DM method. The selection of the CRISP-DM method framework is due to Asyraf & Prasetya (2024a) is a framework that is often used to analyze data in the data mining process. This framework includes six stages, namely business understanding, data understanding, data preparation, modeling, evaluation, and implementation. In addition, this CRISP-DM method is proven by Asyraf & Prasetya (Asyraf & Prasetya, 2024b) that succeeded in building a model that is quite effective for the classification of types of handicraft products. This is in line with the results of this study which applied the CRISP-DM framework successfully in building a puppet classification model, succeeded in building a model that is quite effective with CRISP-DM.

The first stage in the CRISP-DM method is *business understanding*, which aims to understand business aspects, identify existing problems, and determine research objectives. The purpose of the study is to classify puppets using the CNN algorithm of the DenseNet121 architecture with *Canny* feature extraction. The puppet characters used in this study as a dataset class are 4 (four) images of five pandawa characters from the Mahabaratha story named Arjuna, Bima, Nakula & Sadewa and Puntadewa as a classification class.

The second stage is data *understanding*, which aims to understand the data used, including its characteristics and quality. The dataset used in the form of images of puppet characters was

obtained from *website Kaggle* with the name "*Indonesian* Movie *(Traditional Puppet Show)"* with 4 classes used consisting of Arjuna, Bima, Nakula & Sadewa and Puntadewa with a total of 1028 data. At this stage, the process is carried out *Exploratory Data Analysis* EDA is an important part of the data science process, which focuses on early analysis to produce quality data that can be used in later stages (Isa et al., 2023). From the EDA process that has been carried out, there are 106 images with abnormal lighting, blurry images based on sharpness as many as 15 data and blurry images as many as 15 data. Therefore, feature extraction is carried out using edge detection. Edge detection is a technique for identifying the edges of an object in an image by detecting significant changes in brightness levels or the presence of discontinuities. One of the methods that can be used for edge detection is *Canny*. Feature extraction process using the *Canny* It is carried out at the preparatory stage before the process of classifying the image of wayang kulit.

The third stage is data *preparation*, which aims to prepare the data by adjusting the dataset to suit the needs at the modeling stage. At the data stage *preparation*, data cleaning and edge detection are carried out by the *Canny*. Method *Canny* It was chosen because it is one of the effective approaches in image processing to detect and analyze edges, which is useful in a variety of applications such as object recognition, medical analysis, and computer vision (Saputra et al., 2023). Research conducted by Saputra et al., (Saputra et al., 2023) compared the Roberts, Prewitt, and Canny edge detection methods for the classification of student card identification, and found that the Canny method produced the highest accuracy compared to the Roberts and Prewitt methods. In addition, the *Canny* also results in sharper, connected edges compared to other methods (Khoirani & Ariansyah, 2024b). In the next stage, normalize the data, because according to Kurniadi et al., (2021) Normalization on the image is used to scale up the pixel intensity value of the image. So that in this case, it can improve the image of puppets that have low brightness. Furthermore, it is carried out *encoding* Using *one-hot encoding* to convert the category into a binary format which refers to the research of Fajarendra et al., (2024) that converts each category into a binary vector there is only 1 element that is worth 1 and all other elements are worth 0. Next, data augmentation was carried out with a setting that referred to the research of Ningsih et al., (2024) consists of *width shift* and *height shift* is an image that is randomly shifted *horizontal* (width) and vertical (height) up to 20% of the original size to make the model more resistant to variations in image position. *Zoom range* namely the image in-*zoom in* or *zoom out* up to 20% to provide image scale variation. Last *fill mode* i.e. when augmentation produces an empty area, the filling mode *'nearest'* used to fill that empty area with the nearest neighbor pixel value. The last stage is *split* data using a ratio of 80% of data *train* and data *temporary* 20%. 20% data *temporary* divided into 10% of the data *validation* and 10% of data *test* or a ratio of 80:10:10. The ratio of 80:10:10 results in a better model of 70:15:15, because the model has more data to train (Margarita et al., 2024).

After *preparation* which produces good and structured data, then carry out *modelling* using the CNN algorithm with the DenseNet121 architecture. The choice of the DenseNet121 architecture is evidenced by Bello et al., (Bello et al., 2024b) in exploring the impact of *machine learning* Traditional and Approach *deep learning* which is refined on prediction accuracy. Type *transfer learning* which are enhanced such as *EfficientNetB0, ResNet34, VGG16, Inception_v3,* and *DenseNet121*. *DenseNet121* shows superior performance with an accuracy of 87%.

Meanwhile, this study was also successfully applied to the image of puppet puppets and was evaluated using *confusion matrix* that generate value *accuracy, precision, recall* and *f1-score.*

Results of architecture evaluation *DenseNet121* with feature extraction *Canny* Generate value *accuracy* 93%*, precision* 93%*, recall* 93% and *f1-score* 92%. Accuracy resulting from the DenseNet121 architecture with feature extraction *Canny* get high accuracy, because according to Mulyana & Pratama (2023) method *Canny* has the advantage of reducing *noise* Using *filter Gaussian* to improve image quality. In addition, according to Nasdal et al., (2024) High accuracy in detecting edges, especially at weak edges, and adjustable parameter flexibility for different types of imagery and applications. Therefore, it can be concluded that the application of feature extraction *canny* in the DenseNet121 architecture in the leather puppet classification was successfully implemented and obtained high model accuracy.

CONCLUSION

Based on the results of the research on the classification of puppets using Canny feature extraction on the CNN algorithm with DenseNet-121 architecture, it can be concluded that the implementation of this method has succeeded in classifying puppet images based on the classes of Pandawa Lima characters, namely Arjuna, Bima, Nakula & Sadewa, and Puntadewa. This research follows the CRISP-DM framework, covering the stages of business understanding, data understanding, data preparation, modeling, evaluation, and deployment. The dataset used consisted of 1028 puppet image data carried out by the EDA process and data preparation such as Canny feature extraction, normalization, one-hot encoding, augmentation, and split data. In the modeling stage, the DenseNet-121 architecture is used, and the evaluation is carried out with a confusion matrix. This model produces 93% accuracy, 93% precision, 93% recall, and 92% f1-score. For further research development, it is recommended to deploy an Android application to help users recognize puppet characters and expand their knowledge of Indonesian culture. An increase in the number of puppet image datasets and classification of other types of puppets is also recommended.

BIBLIOGRAPHY

Alfiansyah, N. S., & Litanianda, Y. (2024a). Identifikasi Lumpy Skin Disease Menggunakan Tensorflow Dengan Metode Convolutional Neuron Network. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(4), 7330–7336. https://doi.org/10.36040/jati.v8i4.10238

Alfiansyah, N. S., & Litanianda, Y. (2024b). Identifikasi Lumpy Skin Disease Menggunakan Tensorflow Dengan Metode Convolutional Neuron Network. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(4), 7330–7336. https://doi.org/10.36040/jati.v8i4.10238

Asyraf, H., & Prasetya, M. E. (2024a). Implementasi Metode CRISP DM dan Algoritma Decision Tree Untuk Strategi Produksi Kerajinan Tangan pada UMKM A. *Jurnal Media Informatika Budidarma*, *8*(1), 94. https://doi.org/10.30865/mib.v8i1.7050

Asyraf, H., & Prasetya, M. E. (2024b). Implementasi Metode CRISP DM dan Algoritma Decision Tree Untuk Strategi Produksi Kerajinan Tangan pada UMKM A. *Jurnal Media Informatika Budidarma*, *8*(1), 94. https://doi.org/10.30865/mib.v8i1.7050

Bastian, H. (2023). Perancangan Concept Art Karakter "Arjuna" Pada Film Animasi 3d "Mahabarata." *Jurnal Ilmiah Wahana Pendidikan*, *9*(17), 329–339.

Bello, A., Ng, S. C., & Leung, M. F. (2024a). Skin Cancer Classification Using Fine-Tuned Transfer Learning of DENSENET-121. *Applied Sciences (Switzerland)*, *14*(17). https://doi.org/10.3390/app14177707

Bello, A., Ng, S. C., & Leung, M. F. (2024b). Skin Cancer Classification Using Fine-Tuned Transfer Learning of DENSENET-121. *Applied Sciences (Switzerland)*, *14*(17). https://doi.org/10.3390/app14177707

Fadillah, C. (2025). Klasifikasi Jenis Rempah-Rempah Alami Untuk Kecantikan Menggunakan DenseNet121. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *9*(1), 923–929.

Fajarendra, Y. I., Fauzan, Y. R., & Uyun, S. (2024). Klasifikasi Citra Eurosat Menggunakan Algoritma Knn, Decision Tree Dan Random Forest. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(4), 7754–7761. https://doi.org/10.36040/jati.v8i4.10458

Fatmayati, F., Nugraheni, M., Nuraini, R., & Rossi, F. (2023a). Classification of Character Types of Wayang Kulit Using Extreme Learning Machine Algorithm. *Building of Informatics, Technology and Science (BITS)*, *5*(1). https://doi.org/10.47065/bits.v5i1.3568

Fatmayati, F., Nugraheni, M., Nuraini, R., & Rossi, F. (2023b). Classification of Character Types of Wayang Kulit Using Extreme Learning Machine Algorithm. *Building of Informatics, Technology and Science (BITS)*, *5*(1). https://doi.org/10.47065/bits.v5i1.3568

Isa, I. G. T., Zulkarnaini, Novianti, L., Elfaladonna, F., & Agustri, S. (2023). Exploratory Data Analysis (EDA) dalam Dataset Penerimaan Mahasiswa Baru Universitas XYZ Palembang. *Smart Comp: Jurnalnya Orang Pintar Komputer*, *12*(3), 600–609. https://doi.org/10.30591/smartcomp.v12i3.4125

Khoirani, L., & Ariansyah, R. (2024a). *Aplikasi Pengolahan Citra Untuk Peningkatan Deteksi Tepi Melalui Segmentasi Citra*. *2*(3).

Khoirani, L., & Ariansyah, R. (2024b). *Aplikasi Pengolahan Citra Untuk Peningkatan Deteksi Tepi Melalui Segmentasi Citra*. *2*(3).

Kurniadi, B. W., Prasetyo, H., Ahmad, G. L., Wibisono, B. A., & Prasvita, D. S. (2021). Analisis Perbandingan Algoritma SVM dan CNN untuk Klasifikasi Buah. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA) Jakarta-Indonesia*, *September*, 1–11.

Listyalina, L., Hidayahtullah, M. C., Ikhwan, M., & Putri, N. (2021). Identifikasi Sampah Plastik Otomatis Menggunakan Arsitektur Densenet-121. *Science of the Total Environment*, *760*.

Margarita, D., Maulana, H., & Mandyartha, E. P. (2024). Klasifikasi Penyakit Daun Padi Menggunakan Support Vector Machine Berdasarkan Fitur Mendalam (Deep Feature). *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *9*(4), 1–21.

Muhathir, M., Santoso, M. H., & Larasati, D. A. (2021a). Wayang Image Classification Using SVM Method and GLCM Feature Extraction. *Journal of Informatics and Telecommunication Engineering*, *4*(2), 373–382. https://doi.org/10.31289/jite.v4i2.4524

Muhathir, M., Santoso, M. H., & Larasati, D. A. (2021b). Wayang Image Classification Using SVM Method and GLCM Feature Extraction. *Journal of Informatics and Telecommunication Engineering*, *4*(2), 373–382. https://doi.org/10.31289/jite.v4i2.4524

Mulyana, D. I., & Pratama, A. (2023). Optimasi Deteksi Pengenalan Huruf Hijaiyah Dengan Metode Tepi Canny Dan Morfologi. *INTECOMS: Journal of Information Technology and Computer Science*, *6*(2), 717–725. https://doi.org/10.31539/intecoms.v6i2.7663

Mustafid, A., Pamuji, M. M., & Helmiyah, S. (2020). A Comparative Study of Transfer Learning and Fine-Tuning Method on Deep Learning Models for Wayang Dataset Classification. *IJID (International Journal on Informatics for Development)*, *9*(2), 100–110. https://doi.org/10.14421/ijid.2020.09207

Nasdal, D. D., Soeleman, M. A., & Fanani, A. Z. (2024). Analisis Deteksi Tepi pada Citra Kerusakan Bodi Mobil dengan Metode Canny. *Journal of Information System*, *9*(1), 87–93. https://doi.org/10.33633/joins.v9i1.10550

Ningsih, N., Ramadhani, A. D. W. I., & Santoso, D. (2024). Penggunaan Metode Deep Learning untuk Pengembangan Sistem Komunikasi Cerdas bagi Penyandang Disabilitas. *MIND (Multimedia Artificial Intelligent Networking Database) Journal MIND*, *9*(2), 206–219.

Prabowo, D. P., Ullumudin, D. I. I., & Pramunendar, R. A. (2021a). Prototipe Aplikasi Pengenalan Wayang Kulit Menggunakan CNN Berbasis VGG16. *Jurnal Informatika Upgris*, *7*(2), 64–68.

Prabowo, D. P., Ullumudin, D. I. I., & Pramunendar, R. A. (2021b). Prototipe Aplikasi Pengenalan Wayang Kulit Menggunakan CNN Berbasis VGG16. *Jurnal Informatika Upgris*, *7*(2), 64–68.

Raharjo, S. H., & Ningrum, S. U. D. (2024a). Penguatan Pendidikan Karakter pada Generasi Muda Melalui Personifikasi Karakter Pandawa dalam Wayang Kulit. *Jurnal Humanitas Katalisator Perubahan Dan Inovator Pendidikan*, *10*(3), 452–464.

Raharjo, S. H., & Ningrum, S. U. D. (2024b). Penguatan Pendidikan Karakter pada Generasi Muda Melalui Personifikasi Karakter Pandawa dalam Wayang Kulit. *Jurnal Humanitas Katalisator Perubahan Dan Inovator Pendidikan*, *10*(3), 452–464.

Saputra, R. A., Rayadin, M. A., & Febryanti, W. O. I. (2023). Perbadingan Efisiensi Deteksi Tepi Roberts, Prewitt, dan Canny untuk Identifikasi Kartu Mahasiswa. *Jurnal Informatika*, *10*(2), 136–142. https://doi.org/10.31294/inf.v10i2.16726

Sutama, I. W. (2023). Penanaman Nilai-Nilai Karakter Berbasis Storytelling Cerita Keagamaan (Komunikasi Pembelajaran Melalui Cerita Wanaparwa). *SADHARANANIKARANA:Jurnal Ilmiah Komunikasi HinduInstitut Agama Hindu Negeri Gde Pudja Mataram*, *5*(1), 808–823.

Vipul, B., Nitesh, S., Saurabh, A., Saleem, A., Priyanka, B., & Khalid, A. (2021). Indian Sign Language Recognition Using Python. *International Journal of Advanced Trends in Computer Science and Engineering*, *10*(3), 427–434. https://doi.org/10.1007/978-981-33-4367-2_41