Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

# Application of Pattern Recognition Method to Detect East Sumba Fabric Motifs in Lambanapu Using Convolutional Neural Network Method

**Dewiyanti Anapaki[1]*, Pingky Alfa Ray Leo Lede[2]**

[1] *Informatics Engineering Study Program, Wira Wacana Christian University Sumba*
[2] *Informatics Engineering Study Program, Wira Wacana Christian University Sumba*
rambudewy24@gmail.com[1]*, pingky.leo.lede@unkriswina.ac.id [2]

**Abstract**

Traditional woven fabric motifs from East Sumba represent a cultural heritage rich in aesthetic, symbolic, and philosophical values. However, the process of motif identification is still often done manually, which is time-consuming and prone to errors due to subjectivity and limited visual knowledge. This study aims to apply the Convolutional Neural Network (CNN) method to detect and classify four types of East Sumbanese fabric motifs: chicken, bird, crocodile, and horse. The dataset used consists of 200 RGB color images divided equally into training and test data. The CNN architecture used is MobileNetV2 due to its advantages in efficiency and accuracy of visual pattern recognition. To improve model performance and generalization, image augmentation techniques are used. The training process was carried out on the Google Colab platform, while model evaluation was carried out using a confusion matrix and classification reports with precision, recall, and f1-score metrics. The test results showed that the model was able to recognize motifs with varying accuracy in each class, with a total accuracy of more than 70%. These findings indicate that CNN can be an effective solution in supporting cultural preservation through the digitization and classification of traditional fabric motifs from East Sumba.

***Keywords***: *CNN, Pattern Recognition, East Sumba Fabric Motifs, Image Processing, Cultural Preservation.*

## 1. Introduction

Sumba woven fabric is one of the cultural heritage that has a philosophical meaning which contains sacred life values and is upheld by the people of Sumba. Sumba woven fabric consists of various motifs, such as horse motifs, crocodile motifs, chicken motifs, cockatoo motifs, mamuli motifs, traditional house motifs, and various other motifs, each of which has a different philosophical meaning. Generally, Sumba woven fabric consists of two types of fabric, namely hinggi and lao (cloth and sarong). Hinggi is a long cloth and is usually worn by men while lao is a sarong worn by women [1].

A particular problem faced is the lack of knowledge among young people about East Sumba fabric motifs, which can hinder cultural preservation efforts. many young people in East Sumba are less familiar with traditional fabric motifs such as those in Lambanapu due to the lack of exposure to cultural education and the dominance of modern popular culture, so that CNN technology can be a tool to increase awareness and education through digital documentation [2].

The application of CNN to recognize East Sumba fabric motifs in lambanapu is expected to be an efficient and accurate solution in identifying and classifying motifs based on their visual characteristics. This approach also supports the preservation of fabric motifs digitally, increasing the accessibility of cultural information to the wider community, including young people. However, challenges such as motif variation, inconsistent image quality, and limitations of training datasets need to be addressed to ensure successful implementation.

Based on the above background description, this study aims to develop a CNN-based image recognition system that is able to detect East Sumba fabric motifs automatically and accurately. By utilizing pattern recognition technology, this research is expected to contribute to the preservation of East Sumba culture, so the author proposes This research aims to apply a pattern recognition method to detect East Sumba fabric motifs in the image recognition system. Hence pattern recognition technology, specifically through CNN algorithms. With the introduction of this pattern, it is hoped that it will not only increase public awareness of the cultural value of the locality, but also support the creative industry and the preservation of cultural heritage.

## 2. Literature Riview

### 2.1. Pattern Recognition

Pattern recognition is a subfield of artificial intelligence that deals with pattern detection in data. The automatic grouping of numerical and symbolic data, including images, is achieved through pattern recognition (in this case computers) using classification methods that can group and facilitate the identification of an image into a specific class [3]. The objects that humans see are recognizable because the human brain has learned to classify the objects it encounters in nature in order to be able to distinguish one object from another. Computer machines try to mimic the human visual system with the Method *Convolutional Neural Network* For the Introduction of East Sumba Fabric Motif Patterns [4].

### 2.2. CNN (*Convolutional Neural Network*)

The convolution neural network (CNN) method is a type of artificial neural network that has the ability to recognize patterns in images with a fairly high degree of accuracy [5]. Therefore, CNN can be used in the process of classifying patterns of motifs and colors on woven fabrics. CNN consists of several types of layers, including convolution layers, pooling layers, and fully connected layers. CNN has been developed in various studies related to image processing, because CNN has performance that is able to beat other methods that still use manual feature extraction, and CNN is a deep learning method that has the best performance in image recognition *Convolutional Neural Network* (CNN) is a type of neural network commonly used in image data [6].

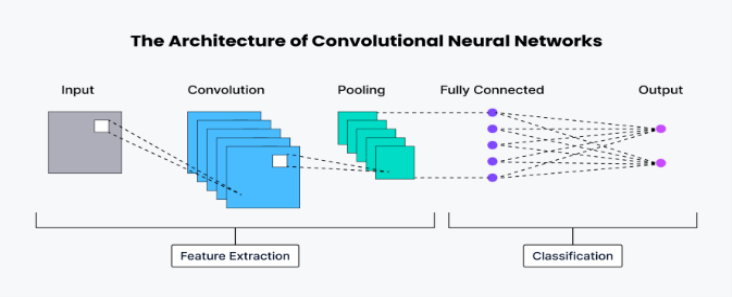


**Figure 1** : CNN architecture

### 2.3. East Sumba Fabric Motif

Customarily and culturally, East Sumba traditional ikat weaving has many functions. In general, this ikat weaving is used as a garment used in dances at traditional parties/ceremonies, as a tool of appreciation and marriage (dowry), as a tool of appreciation and mainstay of death events, as a customary fine to restore disturbed social balance (legal function), as a myth, which is a symbol of a tribe that is glorified because according to certain patterns/designs it will provide protection (from natural disturbances, disasters, evil spirits and others), as well as as a means of appreciation for guests [7] .

### 2.4. Confusion Matrix

The Confusion Matrix is a performance measurement for machine learning classification problems where the output can be two or more classes.

**Table 1** : Matrix Confusion

| class | True Value | | |
|---|---|---|---|
| | | True | Wrong |
| Predictive Value | True | TP (True Positive) | FP (False Positive) |
| | Wrong | FN (FalseNegative) | MR (TrueNegative) |

## 3. Research Methods

The research location was carried out in Lambanapu Village, Lambanapu Village, Kambera District, East Sumba Regency, East Nusa Tenggara Province.

### 3.1. Research Flow

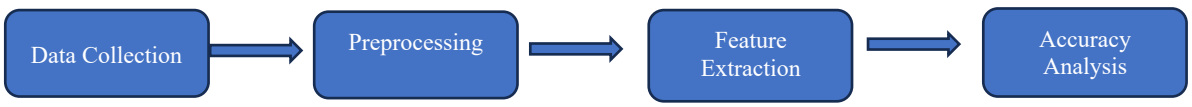


**Figure 2** : Research Flow

### 3.2. Data Collection

This study uses several methods to collect data. First, to obtain relevant information related to the weaving pattern on the fabric, it was collected through interviews with ikat weaving craftsmen, namely: Mr. Tay Hamba Ndima and Mrs. Yuliana Ana Hamu. The second method is documentation, which is used to collect the data needed, such as data on pattern drawings on ikat woven fabrics.

### 3.3. Preprocessing

Preprocessing is the initial stage in image processing before the data is used in the CNN model training process.

These preprocessing steps are important to improve data quality and speed up the training process. The preprocessing process carried out is: Dataset normalization and resize process.

a) Normalizing RGB to Grayscale

Normalization aims to equalize the scale of pixel values so that the CNN model can process data more efficiently and stably, while conversion to grayscale simplifies the color information understated by the image, so that the focus of the analysis is more directed to the structure and shape of the object in the image, such as the presence of parasites, at this stage, the input image measuring 224x224 pixels with multiple motifs will be converted into a single grayscale channel using the average formula.

b) Resize Images
Once the image has been successfully converted from RGB to grayscale, the next step is to resize or resize the image. This process is important for equalizing the dimensions of all input images, so that the CNN model can process them consistently and efficiently.

### 3.4. Feature Extraction

Feature extraction aims to capture the distinctive characteristics of the image, such as geometric patterns, edges, textures, or intensity, which are important representations in the classification process. In this study, feature extraction is carried out automatically by layers in the CNN architecture, namely *Convolutional Layer* and pooling layer.

a) Convolutional Layer
At this stage, the result of the revision will be multiplied by the kernel to get the result of convolution.

b) Pooling Layer

At this stage, the results of the convolution process will be pooled by taking the maximum, where the matix that will be produced is 2 x 2 in size.

c) Fully Connected Layer
At this stage, each input will be further processed by multiplying by the weight of each connection, then added by the bias value. The results of these operations are then used as the basis for generating the final output of the classification system.

### 3.5. Accuracy Analysis

Once the model has finished being built and trained, the next stage is to do an accuracy analysis to find out how accurate the model is in performing the classification. Accuracy is calculated from the comparison between the number of correct predictions and the total amount of test data.

1) Confusional matrix

The model's prediction results are then compared with the actual labels, and arranged in the form of a matrix to determine the number of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). This analysis provides a preliminary overview of the model's performance, including its ability to identify positive and negative cases.

*a.* *Accuracy*
Measure how many predictions are correct compared to the entire amount of data.

*b.* *Accuracy*
Measure what percentage of the prediction is correct from the total positive prediction.

c. Recall
Measure how many positive predictions are found from actually positive.

d. F1 Score
F1-score is a measure of the harmonious average between precision and reccal that is used to provide a balance between the two metrics.

## 4. Results and Discussions

### 4.1 Data Collection

Data collection in this study was carried out directy in Lambanapu Village, Lambanapu Village, Kambera District, East Sumba Regency, East Nusa Tenggara Province. The data collected was in the form of images of East Sumba ikat woven fabric motifs, which were documented directly from traditional woven fabrics in the area. The images include four main types of motifs, namely horse, chicken,

crocodile, and bird motifs, which are representations of the cultural values of the people of East Sumba. The collection of images was carried out through direct documentation methods of woven fabrics that had been made by local artisans, and supported by interviews with local community leaders to ensure accurate classification of motifs. The dataset obtained consisted of 200 images divided equally into four classes of motifs, each with 50 images. All images have gone through a visual validation process to ensure the accuracy of the motif categories. This dataset is used as the main material in the training and testing of Convolutional Neural Network (CNN) models, with the aim of developing an automatic and accurate fabric motif classification system. The main goal of this data collection process is to provide a representative and high-quality dataset that can be used to build a CNN-based visual pattern recognition system, which supports local cultural preservation efforts through digital technology.

## 4.2 Preprocessing

The code is part of the preprocessing process, specifically to convert images from color format (RGB) to grayscale format. First, the image is read using the cv2.imread() function, which by default produces an image in BGR (Blue, Green, Red) format. Therefore, the conversion from BGR to RGB was carried out using cv2.cvtColor(img_bgr, cv2. COLOR_BGR2RGB) to make the drawings conform to the standard color sequence. After that, the RGB image is converted to grayscale using cv2. COLOR_RGB2GRAY. The process of converting to grayscale aims to simplify visual data by taking only light intensity information (without color), thus simplifying the process of analyzing and training the CNN model. Finally, the RGB and grayscale images are displayed side-by-side using matplotlib.pyplot to provide a comparative visualization between the original image and the conversion result. This stage is important before images are used as inputs in machine learning or deep learning models.

**Figure 3** : Preprocessing code

The image above shows the results of the image preprocessing process carried out in this study, namely the conversion from RGB format to grayscale. The image on the left is an original color image (RGB) of the East Sumba fabric motif which contains red, white, blue, and black color information. Meanwhile, the image on the right is the result of a conversion to grayscale format, which only contains light intensity information in grayscale. This conversion process is performed to simplify the visual representation of the image, reduce the complexity of the data, and speed up the CNN model training process without losing important patterns or motif structures. Although the color is removed, the shape and contour of the object remain clearly visible in the grayscale version, so it can still be used as an effective input for the fabric motif classification process. This stage is an important part of preprocessing in a deep learning-based visual pattern recognition system.


**Figure 4** : Preprocessing Results

## 4.3 Feature Extraction

The image above is an implementation of the feature extraction process from a grayscale image using a simple Convolutional Neural Network (CNN) architecture with TensorFlow/Keras. First, the image size (150x150) with 1 channel (grayscale) was determined, then a special CNN model was built to extract features without classification. The model consists of two convolutional layers (Conv2D) and two pooling layers (MaxPooling2D), which serve to take the characteristics of images such as shapes or patterns from fabric motifs. After that, the convolution results are flattened using Flatten to generate a feature vector. To enter the data, use an ImageDataGenerator with rescaling (rescale=1./255) so that the pixel value is in the range of 0–1. The images are then loaded from the /content/dataset/test directory in grayscale mode with batch_size=1, without a label (class_mode=None) because the primary purpose is feature extraction, not classification. The dataset is not randomized (shuffle=False) to keep the sequence of images consistent. The final output of this process is a feature vector from the image that can be used for further analysis, such as classification, clustering, or visualization of feature distribution.

**Figure 5** : Feature Extraction Code

The image below shows the results of feature extraction from grayscale images of East Sumba fabric motifs using the Convolutional Neural Network (CNN) model. At the top is the original image in grayscale format, which is used as an input for the extraction process. The image shows a distinctive chicken motif pattern, with striking shape details. After going through several layers of convolution and pooling in the CNN model, this image was transformed into a high-dimensional feature vector. In this extraction, the resulting feature vector has a total of 82,944 elements, which represent important information from the image such as shapes, textures, and visual patterns. The vector is the result of the flattening process on the final CNN output, and can be used for further classification or analysis. In other words, instead of looking at the entire image directly, the model simply processes this feature vector to recognize or distinguish patterns. This process is crucial in pattern recognition in CNN-based image classification systems.


**Figure 6** : Feature Extraction Results

## 4.5 CNN Model Architecture

The Convolutional Neural Network (CNN) model used in this study was designed to detect and classify East Sumba ikat woven fabric motifs, such as horses, chickens, crocodiles, and birds. Model architecture consists of several main stages, starting with two convolutional layers that serve to extract important visual features from the fabric imagery, such as geometric patterns, shapes, and motif textures. After passing through the convolution process, feature data is compressed using a pooling layer to reduce dimensions and prevent overfitting, without losing important information. The image extracted from the feature is then flattened through the flattening process, and further processed on the fully connected layer as the final classifier. The output of the model is in the form of four prediction classes, each representing one of the motifs: horse, chicken, crocodile or bird. This classification process uses a softmax activation function that results in a probability distribution for each class. With a large number of training parameters and the use of image preprocessing (such as conversion to grayscale and resize), these CNN models have a fairly complex and efficient ability to recognize the distinctive visual variations of each fabric motif. This allows the model to perform automatic and accurate motif classification, as well as support the preservation of local culture through digital technology.

```
# Bangun model CNN
model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(128, 128, 1)),
    MaxPooling2D(2,2),

    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D(2,2),

    Conv2D(128, (3,3), activation='relu'),
    MaxPooling2D(2,2),

    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(4, activation='softmax')  # 4 kelas
])
```

**Figure 7** : CNN Model Code

This CNN model is built sequentially using three convolutional layers (Conv2D) with the number of filters 32, 64, and 128 respectively, which are followed by a MaxPooling2D layer to reduce the dimension of the feature. This convolutional layer serves to extract important features from 128x128 pixel grayscale images. Once the feature is extracted, the data is flattened via Flatten() and then processed by a dense layer with 128 neurons and a ReLU activation function. To reduce overfitting, Dropout(0.5) is used. Finally, the output layer Dense(4, activation='softmax') is used to classify the image into four classes according to the type of fabric motif.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d_4 (Conv2D) | (None, 126, 126, 32) | 320 |
| max_pooling2d_4 (MaxPooling2D) | (None, 63, 63, 32) | 0 |
| conv2d_5 (Conv2D) | (None, 61, 61, 64) | 18,496 |
| max_pooling2d_5 (MaxPooling2D) | (None, 30, 30, 64) | 0 |
| conv2d_6 (Conv2D) | (None, 28, 28, 128) | 73,856 |
| max_pooling2d_6 (MaxPooling2D) | (None, 14, 14, 128) | 0 |
| flatten_2 (Flatten) | (None, 25088) | 0 |
| dropout_2 (Dropout) | (None, 25088) | 0 |
| dense_4 (Dense) | (None, 128) | 3,211,392 |
| dense_5 (Dense) | (None, 4) | 516 |

**Figure 8** : CNN Model Results

## 4.6 Epoch Evaluation

After model complexion, enter this process model training to measure the performance or accuracy of the model that has been trained using test data or lattim data. The goal is to find out how well the model can predict data that has never been seen before. Evaluation is very important so that we know if it is working well, overfiting, underfiting, or needs to be improved.

```
history = model.fit(
    train_generator,
    validation_data=test_generator,
    epochs=epochs,
    callbacks=callbacks,
    verbose=1
)
```

**Figure 9** : Epoch Code

The image shows the training results of CNN models for 10 of the total 30 epochs. Each epoch shows the important metrics used to evaluate the model's performance, namely accuracy, loss, val_accuracy, and val_loss, both in training data and validation data. In the first epoch, the model was still in the early stages of learning with a low accuracy of around 24.27% and a high loss value of 2,362. However, as the number of epochs increases, there is a gradual increase in accuracy and a decrease in loss values, which indicates that the model is beginning to understand patterns in the data. For example, in the 6th epoch, the accuracy reached 41.83% with the loss value dropping to 1.664, and val_accuracy increasing to 32.99% with val_loss 1.4966. This shows that the model has improved performance in both training data and validation data.

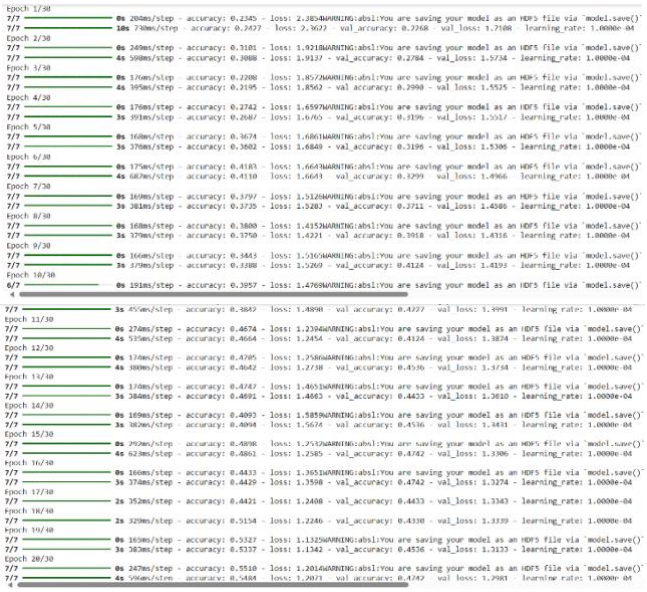

**Figure 10** : Epoch Results

## 4.7 Accuracy and Loss

After getting the results of the evaluation of the model, proceed to determine the accuracy and los. This visualization was created using the matplotlib library. pyplot with a specified image size of 12x5 inches. The first graph shows the accuracy per epoch, both in training data (Train Acc) and validation data (Val Acc). Both are depicted in a single subplot, with the x-axis representing the number of epochs and the y-axis denoting the accuracy value. This graph helps monitor whether the model learns well over time and whether overfitting or underfitting occurs. Furthermore, the second subplot displays the loses (losses) on the training data (Train Loss) and validation (Val Loss) per epoch. Just like accuracy charts, loss charts also use the x-axis as the epoch and the y-axis as the loss value. Loss illustrates how far the model's prediction is from the actual label, so this graph is important for seeing the convergence of the model. Finally, the plt.tight_layout() function is used to set the layout so that it doesn't overlap, and plt.show() displays the entire graph to the screen. This visualization is very useful in evaluating the performance and stability of the model during the training process.

```
# ✅ 9. Visualisasi akurasi dan loss
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Val Accuracy')
plt.title('Akurasi Training vs Validasi')
plt.xlabel('Epoch')
plt.ylabel('Akurasi')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.title('Loss Training vs Validasi')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()

plt.tight_layout()
plt.show()
```

**Figure 11** : Accuracy and Loss Code

The image below shows two graphs illustrating the performance of the Convolutional Neural Network (CNN) model in the training and testing process of East Sumba fabric motif image datasets that have been converted to grayscale format. The first graph (left) shows the comparison between training accuracy and test accuracy, while the second graph (right) shows the comparison of error or loss values between training data and test data. On the accuracy graph, it can be seen that the accuracy of training increases gradually from the beginning to the end of training. Although there was little fluctuation, the trend generally showed a fairly significant increase, with values approaching 63% by the end of the 30th epoch. This shows that the CNN model is starting to learn and recognize patterns in the training data quite well. Meanwhile, accuracy in the test data also improved at the beginning of the training, but tended to stagnate at around 48% after the middle of the epoch. This indicates that although the model learns from training data, its ability to recognize new patterns that has never been trained is still limited.

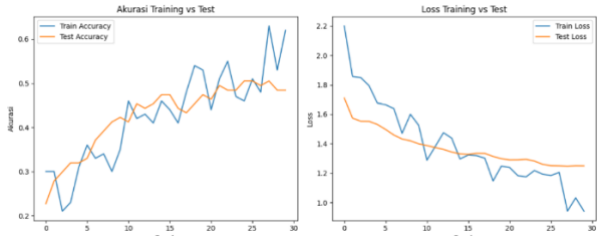


**Figure 12** : Accuracy and Loss Results

## 4.8 Evaluation of the CNN Model

After the Convolutional Neural Network (CNN) model training process is completed, the next step is to evaluate the model's performance on the test data. The code in the image above is used to find out how well the model is in classifying the motif images of East Sumba fabrics based on the four classes used (for example: horses, chickens, birds, crocodiles). The evaluation was carried out with several metrics, namely accuracy, loss value, classification report, and confusion matrix. First of all, the model is tested using test_generator, which is pre-processed test data. The model.evaluate() function generates two main values: loss and accuracy, which reflect how well the model minimizes prediction errors and how often the model makes correct predictions. The accuracy value is multiplied by 100 and is displayed as a percentage, while the loss value is displayed with four numbers after a comma to indicate the model's error rate against the test data. After that, a prediction is made on all test data using the model.predict(). The prediction output (Y_pred) is still in the form of probabilities for each class, so it needs to be changed to an actual class label by using np.argmax().

```
import numpy as np
from sklearn.metrics import classification_report, confusion_matrix

loss, acc = model.evaluate(test_generator)
print(f"Akurasi Data Uji: {acc*100:.2f}%")
print(f"Loss Data Uji: {loss:.4f}")

Y_pred = model.predict(test_generator)
y_pred = np.argmax(Y_pred, axis=1)

print("\n=== Classification Report ===")
print(classification_report(test_generator.classes, y_pred, target_names=test_generator.class_indices.keys()))

print("\n=== Confusion Matrix ===")
print(confusion_matrix(test_generator.classes, y_pred))
```

**Figure 13** : CNN Model Evaluation

The image shown below is a confucion matrix which shows the results of the performance evaluation of the classification model for four classes: "chicken, horse, crocodile, bird.

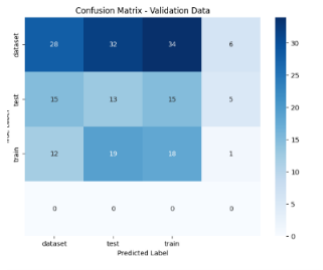


**Figure 14** : Confusion Matrix

## 4.9 Evaluation Matrix

After getting the results of the model evaluation above, proceed to the stage of displaying evaluation metrics such as accuracy, precision, recall, and f1-score for each class, namely chickens, horses, crocodiles, and birds. Classification reports are useful for finding out how well the model detects each class, whether the model tends to be better at recognizing one class than another, and whether there is a balance in classification performance.

```
import numpy as np
from sklearn.metrics import classification_report, confusion_matrix

loss, acc = model.evaluate(test_generator)
print(f"Akurasi Data Uji: {acc*100:.2f}%")
print(f"Loss Data Uji: {loss:.4f}")

Y_pred = model.predict(test_generator)
y_pred = np.argmax(Y_pred, a
                           DirectoryIterator: test_generator
                           DirectoryIterator with 7 items
print("\n=== Classification
print(classification_report(test_generator.classes, y_pred, target_names=test_generator.class_indices.keys()))
```

**Figure 15 :** Evaluation Matrix

The results of the model evaluation below show that the model has quite good classification ability in distinguishing the conditions of "chickens, horses, crocodiles, birds".

```
=== Classification Report ===
              precision    recall  f1-score   support

        ayam       0.22      0.31      0.26        13
       buaya       0.46      0.43      0.44        14
      burung       0.80      0.53      0.64        15
        kuda       0.44      0.47      0.45        15

    accuracy                           0.44        57
   macro avg       0.48      0.43      0.45        57
weighted avg       0.49      0.44      0.46        57
```

**Figure 16** : Evaluation Matrix Results

### 4.10 Implementation

Based on the results displayed on the manual prediction table, the system has tested eight sample images representing four classes each: birds, horses, crocodiles, and chickens. The "Original Label" column shows the actual label of the image, while the "Prediction" column shows the classification results from the model. The confidence value indicates the model's level of confidence in the given predictions. Of the eight images tested, five images were correctly classified according to their original labels (marked with ✅ ), while the other three images were incorrect (marked with ❌ ). This results in a manual accuracy of 62.50%, which means the model is able to recognize fabric motifs with a success rate of about 6 out of 10 images. Although some predictions have been correct, there are still errors, especially in the images of crocodiles and chickens that are sometimes recognized as birds. This indicates that the model still has weaknesses in distinguishing similar visual features between classes, so improvements need to be made such as augmenting more diverse data, increasing the amount of training data, or improving the model architecture.

**Table 2** : Prediction Results

| Yes | File Name | Original Label | Predictions | Cofidence | Information | Explanation |
|---|---|---|---|---|---|---|
| 1. | Birds (1).jpg | Bird | Bird | 0.56 | Yes / ~~No~~ | The prediction is correct – the model recognizes the image of the bird with 56% confidence. |
| 2. | Birds (1) (1).jpeg | Bird | Bird | 0.56 | Yes / ~~No~~ | Predictions are correct – models consistently recognize birds with equal confidence. |
| 3. | Crocodile (1).jpeg | Crocodile | Bird | 0.38 | ~~Yes~~ / No | Wrong prediction – crocodiles were mistaken for birds, possible similar visual features. |
| 4. | Crocodile (19) (5).jpeg | Crocodile | Crocodile | 0.55 | Yes / ~~No~~ | The prediction was correct – the model recognized the crocodile with considerable certainty. |
| 5. | horse (2).jpg | Horse | Bird | 0.46 | ~~Yes~~ / No | Wrong prediction – horses are classified as birds. |
| 6. | horses (13) (2).jpeg | Horse | Horse | 0.47 | Yes / ~~No~~ | The prediction is correct – horses are recognized even though the confidence is only 47%. |
| 7. | chicken (2).jpeg | Chicken | Chicken | 0.38 | Yes / ~~No~~ | The prediction is correct – chickens are recognized despite low confidence. |
| 8. | chicken (43).jpeg | Chicken | Bird | 0.37 | ~~Yes~~ / No | Wrong prediction – chickens are mistaken for birds because of overlapping visual features. |

## 5. Conclusion

This study successfully applied the Convolutional Neural Network (CNN) method based on the MobileNetV2 architecture to detect and classify the motifs of traditional woven fabrics of East Sumba, which consisted of four classes: chicken, bird, crocodile, and horse. The dataset used consisted of 200 RGB images, divided equally into training data and test data (50% each). The training process was carried out using Google Colab, with data augmentation techniques applied to improve the generalization of the model. The test results showed that the CNN model was able to recognize and distinguish each type of motif with varying degrees of accuracy. Performance evaluation using confusion matrix and classification report (precision, recall, and f1-score) showed that the model performed quite well, with an overall accuracy of more than 70%. This proves that the CNN method is effectively used in the process of digitizing and recognizing traditional fabric motif patterns, and can support efforts to preserve local culture through artificial intelligence-based technology. Thus, CNN's approach is not only efficient for motif-based image classification, but also opens up opportunities for the development of automatic recognition systems that can be used by craftsmen, cultural preservationists, and app developers in the textile and cultural industries.

## Reference


[1] A. M. Jawa and A. Iriani, "Knowledge Base of Sumba Woven Fabric Values with Seci Model and Convolutional Neural Network," *Aiti*, vol. 20, no. 1, pp. 1–15, 2023, doi: 10.24246/aiti.v20i1.1-15.

[2] S. Bula, M. Tobu, Y. H. Duka, A. L. Nono, and J. A. Prasetyo, "East Sumba Ikat Weaving: Gender Equality in Cultural Heritage Conservation," vol. 7, no. 2, 2023.

[3] D. C. Khrisne, P. Studi, T. Elektro, F. Teknik, and U. Udayana, "PATTERN RECOGNITION OF GRINGSING WOVEN FABRIC MOTIF PATTERN USING CONVOLUTIONAL NEURAL NETWORK METHOD WITH ALEXNET ARCHITECTURAL MODEL," vol. 6, no. 3, pp. 159–168, 2019.

[4] D. Rainord, L. Gede, I. G. Santi, L. Arida, and A. Rahning, "Implementation of Convolutional Neural Network Method for Android-Based Pattern Recognition of Rote Ndao Woven Fabric Motif Patterns," vol. 11, no. 1, pp. 157–166, 2022.

[5] T. Wahyu, "Implementation of Convolutional Neural Network (CNN) Method for Motif Classification in Sasirangan Imagery," vol. 1, no. 7, pp. 647–653, 2023.

[6] P. Dabbo and F. Y. Bisilisin, "Classification of Raijua Sabu Woven Fabric Motifs Using Image-Based Convolutional Neural Network (CNN)," *TYPE J. Inform.*, vol. 1, no. 06, pp. 11–18, 2024.

[7] R. Rambu Babang and A. Rachmad Rinata, "Marketing Communication Strategy of Prailiu Weaving Center in Increasing Sales of East Sumba Woven Fabrics," *J. Common. Nusant.*, vol. 1, no. 2, pp. 78–85, 2019, doi: 10.33366/jkn.v1i2.24.