East Kalimantan Batik Image Classification Using CNN Architecture with Advanced Feature Extraction

Novianti Puspitasari[1*], Andi Tejawati[2] and Amin Padmo Azam Masa[3]

[1,2]Informatics Study Program, Faculty of Engineering, Mulawarman University, Samarinda, Indonesia
[3]Information System Study Program, Faculty of Engineering, Mulawarman University, Samarinda, Indonesia
Corresponding Author: novipuspitasari@unmul.ac.id

Abstract: Batik is Indonesia's rich and diverse cultural heritage, with various types depending on its origin. One of them is East Kalimantan Batik (Batik Kaltim), which has many batik motifs including Tengkawang Ampiek batik, Batang Garing batik, Kuntul Perak batik, and Mandau batik. Despite its beauty and high artistic value, the public still needs to gain more knowledge about the diversity of East Kalimantan batik, so digitalization is required to introduce these types more widely. This study aims to carry out the image classification process to recognize the kinds of East Kalimantan batik and make an effort to digitize batik. The classification process also uses one Non-East Kalimantan batik motif to differentiate data. Based on this, implementing classification techniques with Convolutional Neural Network (CNN) is very relevant because CNN can analyze batik images and classify them based on existing characteristics. To support feature extraction, the SIFT (Scale-Invariant Feature Transform) and AKAZE (Accelerated-KAZE) methods are applied, which help detect and describe important features in batik images. The research results obtained an accuracy value in the combination of SIFT-CNN methods of 95%, while AKAZE-CNN recorded an accuracy of 89%. Based on the test results, the accuracy obtained performs well using four convolution layers. In addition, combining SIFT and CNN methods is the best approach in classifying East Kalimantan batik.

Keywords: AKAZE Method; Batik Kaltim; Convolutional Neural Network; Feature Extraction; Image Classification; SIFT Method

1. Introduction

Convolutional Neural Network (CNN) is a development of Multi-Layer Perceptron (MLP) specifically designed to process two-dimensional data [1]. CNN is included in the category of deep neural networks because of its deep layer structure, which allows this method to be effective in detecting and recognizing objects in images [2]. CNN is one of the main approaches in object classification in image data and has proven to be an effective solution in overcoming classification challenges in computer vision, especially in the context of digital image recognition [3]. One significant application of CNN is in pattern recognition and classification of batik images, where the pattern identification process is carried out based on the texture contained in the image. Grouping batik patterns often involves analyzing repeating patterns that appear in pixels in the image [4][5].

Batik, found in various regions in Indonesia, has diverse patterns that reflect the richness of culture, philosophy, and local environment. Generally, batik in Indonesia can be grouped into three large categories: Keraton Batik, Coastal Batik, and Interior Batik [6][7]. Several distinctive batik types have emerged in East Kalimantan, or Kaltim, showcasing local traditions and natural motifs. Notable examples include Batik Tengkawang Ampiek, which features the Tengkawang tree and its nuts, symbolizing abundance and life. Batik Batang Garing draws inspiration from local flora and fauna, often incorporating intricate designs representing the region's biodiversity. Batik Kuntul Perak showcases the elegance of the silver heron, a bird associated with grace and beauty. At the same time, Batik Mandau is characterized by patterns that mimic the traditional sword of the Dayak people, reflecting strength and heritage. Given the large number and diversity of types of batik, an effective batik introduction process is

Received: September 30th, 2024. Accepted: June 2nd, 2025
DOI: 10.15676/ijeei.2025.17.2.6

needed to help identify and classify various kinds of batik and provide useful information for the community. One of the main challenges in pattern recognition is classifying images into appropriate categories. Classification can be done in batik objects based on texture [8][9]. Recognizing batik image textures requires detecting boundaries between various textures, which in turn requires deep learning techniques to identify and distinguish batik textures effectively. The classification results can be optimized with the right technique to reflect the existing texture differences [10].

The research that has been conducted includes highlighting the importance of feature selection in batik motif classification, especially the combination of texture and shape features, to improve accuracy. With the information gain value approach, the top ten features were selected to be processed in an artificial neural network. The results showed that the classification accuracy reached 75%, while the application of feature selection increased the accuracy to 87.5%, recording an increase of 12.5%. This finding confirms that the right feature selection greatly affects the effectiveness of batik motif image classification [11]. Another study is about Papuan Batik motifs using four dataset classes: Cendrawasih, Raja Ampat, Tifa Honai, and Asmat, with a deep learning approach through the VGG16 and ResNet50 architectures. The results showed an accuracy of 78.79% for VGG16 and 81.82% for ResNet50 without augmentation. The augmentation technique increased accuracy to 84.85% for VGG16 and 87.88% for ResNet50 [12]. In other research, batik from various regions in Indonesia is introduced [13]. Different from previous research, in this research, in addition to the differences in the methods implemented, the data used is also different, namely in the form of motif images that specifically come from batik cloth from East Kalimantan Province. The introduction of batik in this research focuses on East Kalimantan batik cloth motifs based on the Dayak community's culture and environment. The Dayak batik motifs studied are the Batang Garing, Mandau, Burung Enggang, Shaho, Kuntul Perak, and Tengkawang Ampiek motifs.

In the context of developing a batik image classification system using Convolutional Neural Networks (CNN), CNNs were chosen due to their advantage in image processing over RNNs, LSTMs, and Transformers. This superiority arises from their architecture, specifically tailored to leverage spatial structures via convolutional and pooling layers, which effectively extract hierarchical features, are parameter-efficient through weight sharing and local connections, and exhibit invariance to minor transformations. In addition, the feature extraction process in image processing is a fundamental component influencing the model's overall performance [14]. CNN can automatically extract features from images through convolution layers, but the effectiveness of this process is highly dependent on the quality and characteristics of the resulting features [15]. Feature extraction techniques such as Scale-Invariant Feature Transform (SIFT) and Accelerated-KAZE (AKAZE) offer alternative methods that can enrich the representation of features in images in ways that CNN may not fully cover [16][17]. In the case of batik images, where motifs and patterns have high complexity and diversity, choosing the right feature extraction method can improve the ability of the CNN model to recognize and classify various types of batik patterns more accurately [8]. The importance of the feature extraction process in image classification lies not only in its ability to filter and clarify key information from image data but also in its impact on the efficiency and accuracy of the classification model. By utilizing relevant and significant features, the classification system can more effectively distinguish between image categories, reducing the influence of noise and unimportant variability [18][19]. Although CNN can perform feature extraction automatically, understanding and integrating traditional feature extraction methods such as SIFT and AKAZE can provide additional benefits in optimizing model performance. Therefore, research comparing these feature extraction methods is important to ensure that the built batik image classification system can achieve high accuracy and efficiency [20].

SIFT and AKAZE methods offer significant advantages in image feature extraction [21]. SIFT excels in detecting and describing features at different scales and orientations, effectively dealing with scale changes, rotations, and image shifts. The features generated by SIFT are invariant to these transformations, allowing for reliable feature matching even when the image experiences variations or distortions. This advantage makes SIFT very useful in object recognition and image matching applications that require robustness to shape and orientation changes [22]. AKAZE, on the other hand, offers higher processing speeds than SIFT due to its algorithmic optimizations. A non-linear scale space-based feature detector in AKAZE allows for faster feature detection without sacrificing quality. AKAZE is also more robust to noise and provides stable results on low-quality images. These advantages make AKAZE a good choice for applications that require real-time image processing or in variable lighting conditions [23].

2. Method

A. Research Dataset

This study analyzes the differences between East Kalimantan batik cloth and batik cloth from outside East Kalimantan, namely batik motifs from Yogyakarta. The data used consists of 500 images covering five types of batik with the following details: four types of batik from East Kalimantan, namely Tengkawang Ampiek batik, Batang Garing batik, Kuntul Perak batik, Mandau batik, and one type of batik from outside East Kalimantan, namely Parang Barong Raja Batik from Yogyakarta. The total initial image data used is 500 images, where each type of batik consists of 100 image data. The data will then be divided into training, validation, and test data with a percentage division of 60% training data, 20% validation data, and 20% test data. So the test data used in this study is 20 images per type of batik or 100 images for five kinds of batik. The distribution of this data is designed proportionally to ensure that the developed model can generalize well to all batik analyzed. Figure 1 is an example of the research dataset used.

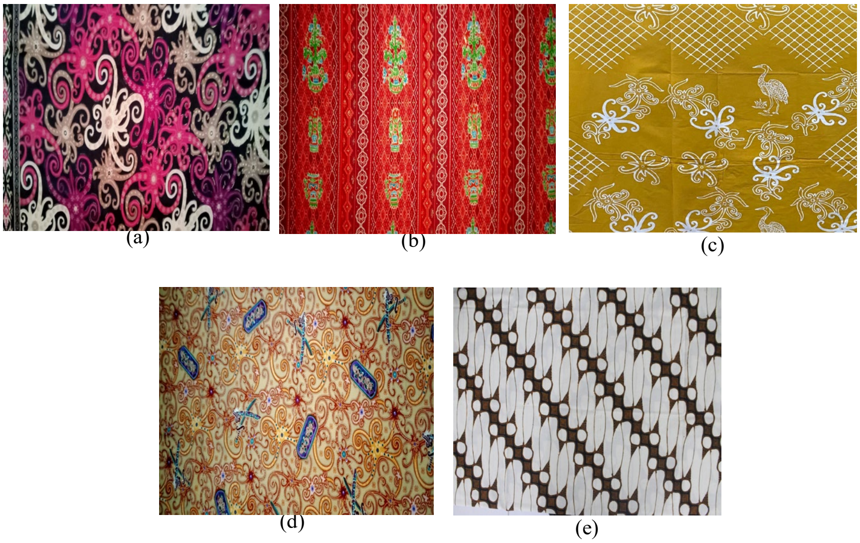


Figure 1. Types of Batik Kaltim (a) Batik Tengkawang Ampiek; (b) Batik Batang Garing; (c) Batik Kuntul Perak; (d) Batik Mandau; and Batik Non Kaltim (e) Batik Parang Barong Raja

B. Research Stages

The stages of this research process begin with image acquisition. After image collection, the next step is preprocessing, where the images are resized to 256×256 pixels to ensure consistency in the analysis [24]. Good preprocessing is essential to obtain accurate results in the next stage. Furthermore, feature extraction is carried out using the SIFT or AKAZE method, where feature descriptors from the image are obtained to describe the unique characteristics of each batik motif. The next process involves training a CNN model using the k-fold cross-validation technique. This method divides the dataset into training and validation subsets for model evaluation, thereby reducing overfitting and increasing the reliability of the results. After the model is trained, the testing stage is carried out by classifying 100 batik images to assess the model's accuracy in identifying the correct class. The classification results are then analyzed to evaluate the effectiveness and accuracy of the two feature extraction methods in the context of batik images. The overall research stages are described in Figure 2.

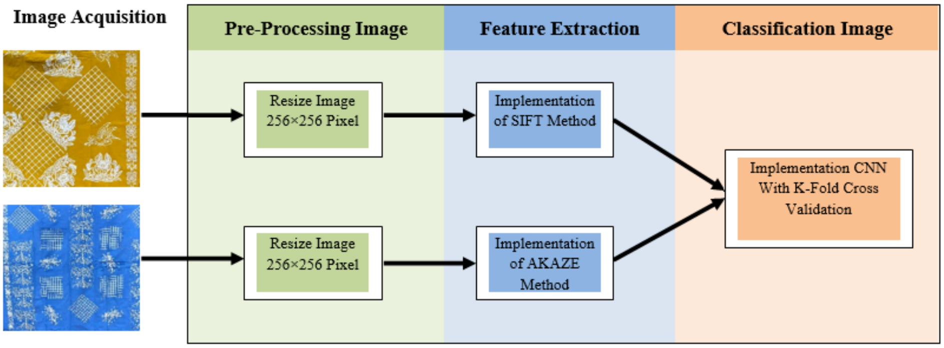


Figure 2. Stages Of The Batik Kaltim Motif Classification Process

C. Pre-processing

Preprocessing is an important initial stage in digital image processing, aiming to prepare the image more suitable for subsequent processing algorithms [25]. One commonly used preprocessing technique is resizing, which changes the size of the image dimensions, either enlarging (upscaling) or reducing (downscaling). At this stage, the original image is changed to 256x256 pixels to ensure consistency and compatibility when used as input in the feature extraction process with the SIFT and AKAZE methods. This size standardization is important to avoid problems arising from differences in image dimensions, thus helping the model recognize significant patterns and increase accuracy in batik image analysis.s of the batik kaltim motif classification process.

D. Feature Extraction

Feature extraction is extracting relevant and distinctive information from data, such as a digital image. These features serve as a numerical representation of an object or region of interest in an image, and good features should have invariant properties, meaning they do not change significantly, even if there are changes in the image, such as rotation, scale, or lighting changes [26]. Feature extraction is very important in various image processing applications, such as object recognition and image matching [27]. The main purpose of feature extraction is to reduce the complexity of image data so that it is easier to process further, such as classification or decision-making [28]. Some popular feature extraction methods include SIFT and AKAZE, each of which has advantages and disadvantages regarding computational speed, accuracy, and resistance to noise. The process flow in the feature extraction process can be seen in Figure 3.

Based on Figure 3, the feature extraction process is carried out after the image preprocessing stage, particularly following the resizing step, to ensure uniform input dimensions. At this stage, two local feature extraction methods SIFT and AKAZE are employed to identify and describe key points within the batik motif images. SIFT detects stable points under scale and rotation transformations, generating high-dimensional descriptors representing the local gradient orientation around each key point. In contrast, AKAZE utilizes a non-linear scale-space approach to detect features with higher computational efficiency, producing more compact binary descriptors. The resulting feature vectors are then transformed into a suitable representation format to serve as input for a Convolutional Neural Network (CNN). The CNN processes this representation during the learning phase to recognize spatial patterns and automatically classify batik motifs. This approach effectively optimizes image representation from local and global spatial perspectives by integrating feature extraction methods such as SIFT or AKAZE with CNN-based feature learning.

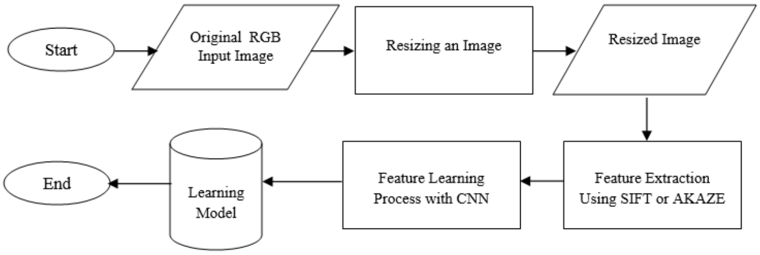


Figure 3. SIFT and AKAZE with CNN Method Implementation Process Flow

1. SIFT Method
   The SIFT method is a feature extraction technique designed to detect and describe key features in images with robustness to scale, rotation, and lighting changes. The SIFT process begins with identifying key points in the image, which is done by detecting local extrema at different scales. This allows SIFT to find relevant features regardless of changes in the size or viewing angle of the image. Once the key points are detected, SIFT produces a feature descriptor that combines information about the pixel intensities around the key points, allowing the model to represent the local characteristics of the image accurately. The advantage of SIFT lies in its ability to produce robust and invariant descriptors, making it very useful in various applications such as object recognition and image matching. SIFT can effectively distinguish objects despite variations in lighting or distortion. The reliability and accuracy of SIFT in detecting key features make it a popular choice in image processing and computer vision, and it is widely used in projects that require in-depth image analysis. The SIFT method has several key equations that play a role in the detection and feature extraction process [29][30]. Here are some important equations in SIFT:
   Extremum Detection: SIFT uses the Laplacian of Gaussian (LoG) to detect key points. The general equation 1 for LoG is:

$$L(x,y,\sigma) = \frac{\partial^2 G(x,y,\sigma)}{\partial x^2} + \frac{\partial^2 G(x,y,\sigma)}{\partial y^2} \tag{1}$$

   where $G(x,y,\sigma)$ is a Gaussian defined as equation 2:

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

   Feature Descriptors: SIFT descriptors are formed from the orientation and intensity of pixels around a key point. They are calculated by dividing the area around the key point into cells and calculating the orientation histogram for each cell.

   Transformation Invariance: SIFT is designed to be invariant to scale and rotation changes. Feature descriptors are calculated in polar coordinates based on the angle and distance from the keypoint.

   Feature Matching: The Euclidean distance is usually used to match descriptors between two images using equation 3, where *a* and *b* are the two descriptors to be matched.

$$d(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{3}$$

2. AKAZE Method
AKAZE (Accelerated KAZE) is a new feature extraction algorithm that offers significant improvements over traditional methods such as SIFT and SURF. AKAZE is designed to be faster and more robust while maintaining the same level of accuracy. It combines the strengths of SIFT and SURF, adopting the scale invariance of SIFT and the computational efficiency of SURF. The AKAZE feature extraction process involves several main stages: keypoint detection, orientation estimation, feature descriptor computation, and feature descriptor matching. The main advantages of AKAZE are its high speed, good accuracy, and robustness to noise, blur, and lighting changes. However, like other algorithms, AKAZE has limitations and continues to be the object of research to improve its performance. AKAZE is a very useful tool in image processing and computer vision. The AKAZE method has several key equations that underlie the feature detection and extraction process. Here are some of the key aspects and equations associated with AKAZE [31]:
Feature Detection with KAZE: AKAZE uses a scale variable-based approach to detect features. The equation to solve feature detection involves using Gaussian Scale Space, expressed by equation 4, where *G(x,y,σ)* is a derived Gaussian function.

$$L(x,y,\sigma) = div(G(x,y,\sigma).\nabla G(x,y,\sigma)) \tag{4}$$

Feature Descriptors: Descriptors in AKAZE are calculated using orientation histograms, similar to SIFT but more efficient. Using a faster calculation approach, they are formed based on the orientation and intensity of pixels around the key point.
Divergence: In AKAZE, the divergence function speeds up the feature extraction process. The gradient's divergence is formulated in equation 5, where *v* is the gradient vector.

$$div(v) = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \tag{5}$$

Feature Matching: Descriptor matching uses Euclidean distance, similar to equation 6, where *a* and *b* are the two descriptors being compared.

$$d(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{6}$$

In the feature extraction stage, the process is carried out by applying the SIFT and AKAZE methods to compare the performance of the two techniques in detecting and extracting features from preprocessed images. The SIFT method focuses on detecting key points resistant to scale and rotation changes, producing robust and representative feature descriptors of the local characteristics of each end, allowing the model to recognize important patterns in the image with high accuracy. Furthermore, AKAZE is applied to explore the efficiency of feature extraction. Although designed to be faster, AKAZE still maintains the ability to detect significant features, especially in complex image conditions. By comparing the results of these two methods, this study aims to determine which method provides better feature representation for subsequent image classification. The following are the results of feature extraction using the SIFT and AKAZE methods in Figure 4.

(a)

(b)

Figure 4. Feature Extraction Process Results (a) SIFT Method; (b) AKAZE Method

In the SIFT and AKAZE methods, the resulting feature values include several important parameters that describe the characteristics of key points detected in the image. The number of key points indicates the number of key points successfully identified, reflecting the complexity and detail of the image. Each key point has coordinates (*x, y*) indicating its specific location in the image and a size indicating the scale at which the feature was detected, providing information about the object's dimensions. Feature descriptors, numerical representations of local characteristics around each key point, provide information about orientation, intensity, and texture patterns. SIFT produces a descriptor with a length of 128 elements, including information about the size, orientation, and response of each keypoint, where size refers to the diameter of the keypoint, orientation describes the rotation angle, and response refers to the strength of the detected keypoint.

The AKAZE method produces a descriptor with a length of 61 elements. Although shorter, this descriptor also stores similar information, including the keypoint's size, orientation, and response. The advantage of AKAZE lies in its efficiency in calculating descriptors, making it faster in feature extraction, especially on larger images. By understanding these parameters, SIFT and AKAZE can be used effectively for object recognition and image matching in various image processing applications. Table 1 and Table 2 is an example of the results of feature extraction values in each SIFT and AKAZE method.

Table 1. The Result of the Feature Extraction SIFT Method

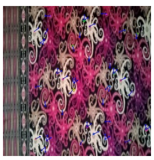
Image	**Number of Keypoints**	**Shape of Descriptors**	**Coordinates**	**Size**	**Orientation**	**Response**
	1260	(1260, 128)	(2.5392, 196.5334)	2.6557	17.9845	0.0147
	872	(872, 128)	(2.7076, 123.8339)	2.3642	288.0860	0.0335

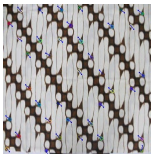
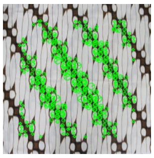
Image	Number of Keypoints	Shape of Descriptors	Coordinates	Size	Orientation	Response
	1372	(1372, 128)	(2.4163, 193.0972)	3.1633	316.3655	0.0332
	1280	(1280, 128)	(2.8443, 150.1296)	3.1820	354.4043	0.0162
	611	(611, 128)	(4.5279, 242.2174)	2.0727	170.4078	0.0532

Table 2. The Result of the Feature Extraction AKAZE Method

Image	Number of Keypoints	Shape of Descriptors	Coordinates	Size	Orientation	Response
	391	(391, 61)	57.6316, 29.1804	4.80	150.0290	0.0010
	92	(92, 61)	(52.1234, 35.7954)	4.80	89.8566	0.0011
	99	(99, 61)	(102.0002, 28.5404)	4.80	67.2599	0.0050
	103	(103, 61)	(180.1150, 32.7114)	4.80	58.9232	0.0012

Image	Number of Keypoints	Shape of Descriptors	Coordinates	Size	Orientation	Response
	366	(366, 61)	(192.6501, 29.5968)	4.80	157.9186	0.0049

Based on Table 1 and Table 2, the differences in characteristics between SIFT and AKAZE can be observed across several key aspects of the feature extraction process. Regarding the number of key points, SIFT tends to generate more than AKAZE due to its multi-scale approach, which is more sensitive to local variations. In contrast, AKAZE is more selective, producing fewer key points but with greater computational efficiency. Regarding keypoint size, SIFT can detect features across a wider scale range, while AKAZE tends to make smaller and more uniform sizes. Both methods compute the orientation of key points in degrees to maintain robustness against rotation, although, in SIFT, orientation is used more explicitly during descriptor construction. As for the response value, SIFT generally yields higher scores due to its Gaussian difference-based detection. AKAZE, which employs a non-linear approach, produces lower values still relevant for feature selection [32][33].

The response value in key point detection represents the strength or distinctiveness of a feature and is computed differently depending on the method used. In SIFT, the response is derived from the Difference of Gaussian (DoG) function. Points are detected as local extrema in the DoG scale space, and the response is typically defined as the absolute value of the DoG function at the given position and scale [34]:

$$Response_{SIFT} \approx |D(x,y,\sigma)| \tag{7}$$

Based on equation 7, where $D(x,y,\sigma)$ is the DoG value at pixel location (x,y) and scale σ. A higher response indicates a more prominent key point regarding local contrast. In AKAZE, the response is calculated using a modified FAST (Features from Accelerated Segment Test) detector in a non-linear scale space. The response reflects the contrast between the center pixel *Ic* and its surrounding pixels *Ii* and is often computed as [34]:

$$Response_{AKAZE} = \sum_{i=0}^{n} |I_i - I_C| \tag{8}$$

or in some cases:

$$Response_{AKAZE} = \min(I_p - I_C) \tag{9}$$

Based on equation 8 where n is the number of sampled pixels in the circular neighborhood, and equation 9 where I_p is a surrounding pixel intensity. The resulting value measures how strongly a pixel differs from its neighborhood, which indicates its salience as a feature. Overall, SIFT often yields higher response values due to its Gaussian-based contrast detection, while AKAZE provides lower but computationally more efficient responses [34].

E. Image Classification

CNN is a special architecture that is adapted and used to classify images. The learning process can quickly be carried out on one convolution layer to be added to several useful layers for optimal learning. CNN uses three basic ideas: local receptive fields, shared weight, and pooling. CNN is simply a sequence of layers that change an activation volume into another form through a differentiable function. The sequence of layers in a CNN is merely a convolutional layer, pooling layer, and fully connected layer. The CNN network architecture shown in Figure 10 explains in broad outline that it is divided into two parts: feature learning (feature extraction layer) and classification (fully-connected layer). CNN is a special architecture that is adapted and used to classify images. The learning process can quickly be carried out on one convolution layer to be added to several useful layers for optimal learning. CNN uses three basic ideas: local receptive fields, shared weight, and pooling. CNN is simply a sequence of layers that transforms an activation volume into another form through a differentiable function. The sequence of layers in a Convolutional Neural Network is merely a convolutional layer, a pooling layer, and a fully-connected layer [35][36]. The CNN network architecture shown in Figure 5 explains that it is broadly divided into two parts, namely feature learning (feature extraction layer) and classification (fully connected layer).

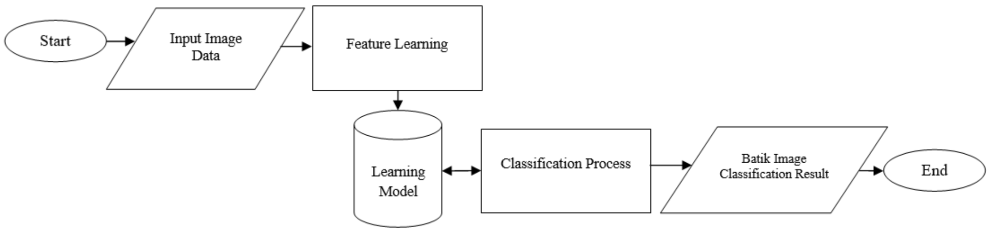


Figure 5. CNN Architecture Process Flow

Accuracy measurement in CNN compares the model's prediction results to the correct labels from the testing dataset. Mathematically, accuracy is calculated as the ratio between the number of correct predictions and the total number of samples Based on equation 7. This accuracy measurement is important to evaluate the model's performance and ensure that the network can recognize patterns well in data that has never been seen before. The equation provides the percentage of correct predictions, reflecting how well the model classifies the data. The higher the accuracy value, the more effectively the model recognizes patterns in the testing data. This method is a common and direct way to evaluate the performance of a classification model, providing a clear picture of the algorithm's success.

$$Accuracy = \frac{Total\ number\ of\ testing\ data}{Number\ of\ correct\ predictions} \times 100\% \tag{7}$$

3. Results and Discussion

Based on the test scenario, a comparison of the accuracy values of the classification process was obtained by combining the use of the SIFT-CNN method with AKAZE-CNN. The study's results present clear and structured data in tables that allow for an easier analysis process. Each accuracy value obtained reflects the effectiveness of the feature extraction method applied, either using SIFT or AKAZE. The feature extraction process of each SIFT and AKAZE method will be used as input for the image recognition process with CNN. In the test scenario, the search for accuracy values is carried out by comparing the accuracy values to the depth or number of convolution layers on the CNN used. Using various convolution layers provides insight into how model complexity can affect its ability to recognize patterns. Each additional layer can contribute to the model's ability to learn deeper features but also carries the risk of overfitting if too many layers are used without enough training data. By varying the number of convolution layers, this study attempts to determine the optimal configuration that produces the highest accuracy based on equation 7. Table 3 compares the accuracy results using the number of layers in the feature learning process.

Table 3. Accuracy Value In the Training Process

Number of Learning Feature layers	Accuracy Results SIFT-CNN	Accuracy Results AKAZE-CNN
2 layers	65,55%	53,48%
3 layers	68,04%	61,78%
4 layers	93,70%	79,63%
5 layers	88,92%	77,26%
6 layers	87,57%	72,47%

Table 3 shows the effectiveness of combining methods in increasing training accuracy based on the tests conducted. The SIFT method combined with CNN produces the highest accuracy value of 93,70% using four convolutional layers. Meanwhile, combining the AKAZE and CNN methods obtained a training accuracy value of 79,63% using five convolutional layers. The effect of accuracy values on training and testing accuracy is important. Suppose the training accuracy value is high, but the testing accuracy is low. In that case, this may indicate overfitting, where the model learns too much detail on the training data so that it cannot generalize to new data. Conversely, low accuracy values may indicate underfitting, where the model is too simple and cannot capture patterns in the data. Therefore, the balance between training and testing accuracy must be considered to ensure that the model is accurate in learning the training data and effective in applying that knowledge to data that has never been seen before.

Table 4. Accuracy Value in the Testing Process

Number of Learning Feature layers	Accuracy Results SIFT-CNN	Accuracy Results AKAZE-CNN
2 layers	66%	62%
3 layers	67%	66%
4 layers	**95%**	**89%**
5 layers	91%	82%
6 layers	91%	74%

The model prediction results are compared to the correct labels from the testing dataset in the classification process. The SIFT-CNN method obtained good accuracy values from using four layers to six layers in a row: 95%, 91%, and 91%, with the best accuracy value of 95% when using four layers. Meanwhile, the combination of the AKAZE-CNN method recorded good accuracy values starting from using four layers to six layers in a row: 89%, 82%, and 74%, with the best accuracy value of 89% when using four layers. From the test results table, it can be concluded that the research that has been done shows that obtaining prediction results with optimal accuracy values uses four convolution layers. Then, based on Table 4, it can be concluded that the accuracy value of the SIFT-CNN method combination is slightly better than that of the AKAZE-CNN method combination, with a difference of 6% in the testing process.

Based on Table 4, the difference in accuracy between the SIFT+CNN and AKAZE+CNN combinations can be explained by the quality of the feature representation produced by each extraction method. SIFT produces high-dimensional gradient-based descriptors rich in spatial information and robust to geometric transformations such as rotation and scale variations. This allows CNN to learn more complex and accurately localized patterns. In contrast, AKAZE uses binary descriptors that are computationally more efficient but less informative so that the spatial information passed to CNN is more limited. In addition, the distribution of key points produced by AKAZE tends to be more selective and dense in certain areas, compared to SIFT, which makes more key points and is evenly distributed throughout the image area. Thus, CNN, which receives input from SIFT, has an advantage in learning spatial representations, which empirically produces higher accuracy.

Meanwhile, the effect of the number of CNN layers on accuracy shows an optimal point in the architecture with four convolutional layers. In this condition, the network is deep enough to capture complex feature representations, as well as to maintain gradient stability and avoid overfitting. Increasing the number of layers to five or six causes a decrease in accuracy. This is most likely due to overfitting. In addition, the deeper the network, the higher the risk of losing important information due to the vanishing gradient phenomenon or excessive representation compression. Therefore, the balance between network depth and data complexity is crucial for learning success, and a configuration with four layers has been shown to provide the most optimal performance in this context.

4. Conclusion

The conclusion of this study shows that the combination of SIFT-CNN methods produces slightly better accuracy than AKAZE-CNN, with a difference of 6% in the testing process. The results show that four convolutional layers are needed to achieve optimal accuracy, with SIFT-CNN achieving the highest accuracy of 95% and AKAZE-CNN reaching 89%. This study emphasizes the importance of the feature extraction process in image recognition and how varying the number of convolutional layers can affect the model's ability to recognize patterns. The balance between training and testing accuracy is crucial because the model must generalize well without overfitting or underfitting. Thus, choosing the right layer configuration is key to achieving optimal results in image classification.

Future research should explore integrating deep feature extraction techniques, such as learned feature descriptors from pre-trained models (e.g., VGG, ResNet), with traditional local feature methods like SIFT or AKAZE. In addition, replacing SIFT or AKAZE with more recent or learning-based key point detection methods such as Super Point, ORB, or D2-Net may improve feature robustness and computational efficiency. Furthermore, testing the model on larger and more diverse datasets or implementing hybrid architectures with residual or attention mechanisms could enhance generalization and classification performance across varied image domains.

5. Acknowledgment

The author would like to express his deepest gratitude to the Ministry of Education, Culture, Research and Technology of the Republic of Indonesia for providing financial support for the BIMA research project, a fundamental research scheme, based on decree number 0667/E5/AL.04/2024 with contract number: 082/E5/PG.02.00/PM.BARU/2024 and 639/UN/17.L1/HK/2024.

6. References

[1] V. Sinitsin, O. Ibryaeva, V. Sakovskaya, and V. Eremeeva, “Intelligent bearing fault diagnosis method combining mixed input and hybrid CNN-MLP model,” *Mech. Syst. Signal Process.*, vol. 180, p. 109454, 2022.

[2] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artif. Intell. Rev.*, vol. 53, pp. 5455–5516, 2020.

[3] T. Turay and T. Vladimirova, “Toward performing image classification and object detection with convolutional neural networks in autonomous driving systems: A survey,” *IEEE Access*, vol. 10, pp. 14076–14119, 2022.

[4] R. Munir and J. Santoso, “Image Retrieval based on the Fusion of Graph Method with Color Moments, GLCM, and Hu Moments.,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 13, no. 3, 2023.

[5] C. U. Khasanah, E. Utami, and S. Raharjo, "Implementation of data augmentation using convolutional neural network for batik classification," in *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, IEEE, 2020, pp. 1–5.

[6] R. Rahmidani and D. Susanti, "Tanah Liek Batik's Industry in West Sumatra (a Study of Development Problems)," in *3rd International Conference on Accounting, Management and Economics 2018 (ICAME 2018)*, Atlantis Press, 2019, pp. 228–236.

[7] I. Sulthonuddin and H. Herdiansyah, "Sustainability of Batik wastewater quality management strategies: analytical hierarchy process," *Appl. Water Sci.*, vol. 11, pp. 1–12, 2021.

[8] A. A. Kasim, R. Wardoyo, and A. Harjoko, "Batik Classification with Artificial Neural Network Based on Texture-Shape Feature of Main Ornament," *Int. J. Intell. Syst. Appl.*, vol. 9, no. 6, pp. 55–65, 2017, doi: 10.5815/ijisa.2017.06.06.

[9] A. H. Rangkuti, A. Harjoko, and A. Putra, "A novel reliable approach for image batik classification that invariant with scale and rotation using MU2ECS-LBP algorithm," *Procedia Comput. Sci.*, vol. 179, pp. 863–870, 2021.

[10] E. Winarno, A. Septiarini, W. Hadikurniawati, and H. Hamdani, "The Hybrid Features and Supervised Learning for Batik Pattern Classification," *ACM J. Comput. Cult. Herit.*, vol. 17, no. 2, pp. 1–13, 2024.

[11] A. A. Kasim, R. Wardoyo, and A. Harjoko, "The selection feature for batik motif classification with information gain value," in *Soft Computing in Data Science: Third International Conference, SCDS 2017, Yogyakarta, Indonesia, November 27–28, 2017, Proceedings 3*, Springer, 2017, pp. 106–115.

[12] S. Aras and A. Setyanto, "Classification of Papuan Batik Motifs Using Deep Learning and Data Augmentation," in *2022 4th International Conference on Cybernetics and Intelligent System (ICORIS)*, IEEE, 2022, pp. 1–5.

[13] D. G. T. Meranggi, N. Yudistira, and Y. A. Sari, "Batik classification using convolutional neural network with data improvements," *JOIV Int. J. Informatics Vis.*, vol. 6, no. 1, pp. 6–11, 2022.

[14] W. H. L. Pinaya, S. Vieira, R. Garcia-Dias, and A. Mechelli, "Convolutional neural networks," in *Machine learning*, Elsevier, 2020, pp. 173–191.

[15] R. Vaddi and P. Manoharan, "Hyperspectral image classification using CNN with spectral and spatial features integration," *Infrared Phys. Technol.*, vol. 107, p. 103296, 2020.

[16] N. Kaur, N. Jindal, and K. Singh, "An improved approach for single and multiple copy-move forgery detection and localization in digital images," *Multimed. Tools Appl.*, vol. 81, no. 27, pp. 38817–38847, 2022.

[17] X. H. Zhou and Q. J. Shi, "Multiple copy-move forgery detection based on density clustering," *Pattern Recognit. Image Anal.*, vol. 31, pp. 109–116, 2021.

[18] H. Yu, L. T. Yang, Q. Zhang, D. Armstrong, and M. J. Deen, "Convolutional neural networks for medical image analysis: state-of-the-art, comparisons, improvement and perspectives," *Neurocomputing*, vol. 444, pp. 92–110, 2021.

[19] J. Wang, H. Zhu, S.-H. Wang, and Y.-D. Zhang, "A review of deep learning on medical image analysis," *Mob. Networks Appl.*, vol. 26, no. 1, pp. 351–380, 2021.

[20] S. Kumar, B. Janet, and S. Neelakantan, "Identification of malware families using stacking of textural features and machine learning," *Expert Syst. Appl.*, vol. 208, p. 118073, 2022.

[21] S. K. Sharma and K. Jain, "Image stitching using AKAZE features," *J. Indian Soc. Remote Sens.*, vol. 48, no. 10, pp. 1389–1401, 2020.

[22] R. Pautrat, V. Larsson, M. R. Oswald, and M. Pollefeys, “Online invariance selection for local feature descriptors,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, Springer, 2020, pp. 707–724.

[23] O. Yakovleva and K. Nikolaieva, “Research of descriptor based image normalization and comparative analysis of SURF, SIFT, BRISK, ORB, KAZE, AKAZE descriptors,” *Adv. Inf. Syst.*, vol. 4, no. 4, pp. 89–101, 2020.

[24] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, “Image super-resolution via iterative refinement,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4713–4726, 2022.

[25] R. Sarki, K. Ahmed, H. Wang, Y. Zhang, J. Ma, and K. Wang, “Image preprocessing in classification and identification of diabetic eye diseases,” *Data Sci. Eng.*, vol. 6, no. 4, pp. 455–471, 2021.

[26] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, “A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction,” *J. Appl. Sci. Technol. Trends*, vol. 1, no. 1, pp. 56–70, 2020.

[27] M. Bansal, M. Kumar, and M. Kumar, “2D object recognition techniques: state-of-the-art work,” *Arch. Comput. Methods Eng.*, vol. 28, no. 3, pp. 1147–1161, 2021.

[28] Y. Liu, H. Pu, and D.-W. Sun, “Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices,” *Trends Food Sci. Technol.*, vol. 113, pp. 193–204, 2021.

[29] Y. Liu *et al.*, “Improved feature point pair purification algorithm based on SIFT during endoscope image stitching,” *Front. Neurorobot.*, vol. 16, p. 840594, 2022.

[30] Y. Wang, Y. Yuan, and Z. Lei, “Fast SIFT feature matching algorithm based on geometric transformation,” *IEEE Access*, vol. 8, pp. 88133–88140, 2020.

[31] A. Couturier and M. A. Akhloufi, “A review on absolute visual localization for UAV,” *Rob. Auton. Syst.*, vol. 135, p. 103666, 2021.

[32] S. A. K. Tareen and Z. Saleem, “A comparative analysis of sift, surf, kaze, akaze, orb, and brisk,” in *2018 International conference on computing, mathematics and engineering technologies (iCoMET)*, IEEE, 2018, pp. 1–10.

[33] Z. Hossein-Nejad, H. Agahi, and A. Mahmoodzadeh, “Image matching based on the adaptive redundant keypoint elimination method in the SIFT algorithm,” *Pattern Anal. Appl.*, vol. 24, no. 2, pp. 669–683, 2021.

[34] D. Bojanić, K. Bartol, T. Pribanić, T. Petković, Y. D. Donoso, and J. S. Mas, “On the comparison of classic and deep keypoint detector and descriptor methods,” in *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, IEEE, 2019, pp. 64–69.

[35] D. Bhatt *et al.*, “CNN variants for computer vision: History, architecture, application, challenges and future scope,” *Electronics*, vol. 10, no. 20, p. 2470, 2021.

[36] J. Raitoharju, “Convolutional neural networks,” in *Deep learning for robot perception and cognition*, Elsevier, 2022, pp. 35–69.

Novianti Puspitasari received the B.Sc. degree in informatics engineering from the Universitas Islam Indonesia, and the M.Eng. degree in information technology from the Gadjah Mada University, Indonesia. She is currently a lecturer at the Department of Informatics, Mulawarman University. Her research interest is in data science and analytics, artificial intelligence, and machine learning areas. She can be contacted at email: novipuspitasari@unmul.ac.id.

Andi Tejawati is a lecturer at the at the Department of Informatics, Mulawarman University. He is attached to the Computing and Informatics Institutions Indonesia (APTIKOM) societies. Her research interests include information system and artificial intelligence. She can be contacted at email: tejawatiandi@gmail.com.

Amin Padmo Azam Masa received the M.Cs degree in Computer Science from the Gadjah Mada University, Indonesia. He is currently a lecturer at the Information System, Mulawarman University. His research interest is in image processing, artificial intelligence, big data and machine learning areas. He can be contacted at email: aminpadmo@unmul.ac.id.