Received November 7th 2025; accepted December 19th 2025. Date of publication December 31st 2025
Digital Object Identifier: https://doi.org/10/25047/jtit.v12i2.453

# Digital Image-Based Chili Quality Detection Using a Web-Based Convolutional Neural Network

JAJANG JAYA PURNAMA[1], SRI RAHAYU[2], RIDWANSYAH[3], VERRY RIYANTO[4]

[1,2,3,4]Universitas Bina Sarana Informatika, Jakarta, Indonesia

CORESPONDING AUTHOR: JAJANG JAYA PURNAMA (email:jajang.jjp@bsi.ac.id)

**ABSTRACT** Chili is one of the main horticultural commodities in Indonesia, with high economic value and stable market demand. Accurate determination of chili quality levels is an important factor in maintaining quality, selling price, and distribution efficiency. Until now, the process of assessing chili quality has generally been carried out manually through direct visual inspection by experts or field officers. This traditional approach has limitations, such as varying levels of accuracy due to assessor subjectivity and the limited availability of experts. Advancements in digital image processing technology, particularly deep learning, offer opportunities to develop more accurate and consistent automated detection systems. This study proposes a Convolutional Neural Network (CNN) model to classify chili quality levels based on digital images, which is then integrated into a web-based application. This study uses a dataset of 405 chili images from 11 varietal categories, each labeled with quality (good, pest-infested, or unknown), which undergoes preprocessing stages including resizing, normalization, and data augmentation. The CNN model was designed with convolutional layers, max-pooling, dense layers, and a Softmax activation function, and was trained using the Adam optimizer and Categorical Cross-Entropy Loss. The web application implementation was carried out using the Flask framework, allowing users to upload images and obtain prediction results in real time. The testing results showed that the developed CNN model achieved an accuracy of 1.000 on the test data, with reliable detection performance under variations in lighting and image backgrounds. This research contributes to the development of smart agriculture technology by providing an accurate, fast, and easily accessible solution for chili quality detection.

**KEYWORDS**: keyword consist of 3-5 words

## I.INTRODUCTION

In the post-harvest activities of agricultural commodities, accurately determining both the quality and quantity of chili is an essential and decisive factor that significantly influences multiple aspects of the agricultural supply chain. It plays a major role in ensuring the smooth and efficient distribution of produce from farmers to markets, maintaining consistency and stability in supply chains, optimizing storage and transportation processes, and establishing a fair, competitive, and sustainable selling price for the chili in the marketplace. Furthermore, precise assessment of chili quality and quantity helps reduce potential economic losses, supports informed decision-making for traders and distributors, and contributes to meeting consumer expectations for freshness and quality [1][2]. Chili is one of the main horticultural commodities in Indonesia, recognized for its high economic value and stable market demand, and it plays a crucial role in both domestic consumption and the agricultural economy. Beyond being a staple ingredient in Indonesian cuisine, chili supports the livelihoods of countless farmers, traders, and small-scale agribusinesses across the country. Its stable demand throughout the year makes chili an important factor in ensuring food security, price stability, and income generation for rural communities. Furthermore, chili production and distribution contribute significantly to regional agricultural development, stimulate related industries such as food processing and logistics, and reflect the cultural importance of spicy flavors in Indonesian culinary traditions [3][4]. The complex visual characteristics of chili, such as variations in shape, size, and color changes corresponding to quality levels, are important factors in determining its quality and selling price [5]. Chilies that are pest-infested or considered low-quality are those showing physical damage caused by pests such as insects or diseases. Characteristics of pest- or disease-infected chilies include holes or scratches on the fruit skin, color changes such as blackening or yellowing, as well as fungal spots or bacterial stains on the surface. In addition, chilies that are soft, wrinkled, or have degraded taste and texture are also categorized as poor-quality chilies. Such damage reduces their quality and freshness, thereby affecting their market value. Accurate determination of chili quality levels is essential, as errors in identification can lead to price reductions and affect distribution efficiency [6]. Until now, chili quality assessment has mostly been carried out manually by human labor, which tends to be subjective and less consistent, especially when production volumes are high and lighting conditions vary [7].

In traditional practice, the process of diagnosing or assessing chili quality often relies on direct visual observation in the field by experts or inspection officers [8]. This approach has several limitations, such as varying accuracy due to assessor subjectivity and the limited number of experts available to conduct large-scale inspections [9]. This condition has the potential to cause inconsistent results and slow down decision-making processes, especially in supply chains with high production demands [10].

Advancements in digital image processing technology have provided new solutions in agriculture, particularly for the identification and classification of horticultural commodities [11]. One of the studies in image processing was conducted by combining the Canny method with a feed-forward neural network to identify chili varieties [12]. Recent research used the Canny and Sobel edge detection methods to detect chili contours, focusing on analyzing the influence of threshold parameters on edge detection quality. The results of the study showed that selecting appropriate threshold parameters can improve the clarity of object edges and reduce noise [13]. Nevertheless, edge detection-based approaches are still limited to contour extraction and are therefore unable to automatically classify quality levels. Furthermore, that study did not leverage the advantages of CNNs in automatic feature extraction.

In the past five years, deep learning methods, particularly Convolutional Neural Networks (CNNs), have been widely used for image-based object detection and classification [14][15][16]. One study utilized a CNN with DenseNet169 transfer learning to detect the ripeness level of Katokkon chili and achieved high accuracy; however, the study had not yet integrated the model into a web-based application that would facilitate field use [17]. Another study applied YOLOv5 to detect red chilies directly from plant images, but its focus was solely on red chilies and limited to object detection without detailed quality classification across various chili types [18][19]. This indicates that previous studies still have opportunities for further development in the aspect of chili quality classification.

Based on these developments, this study proposes the development of a CNN model to directly detect chili quality levels from digital images, which is then integrated into a web-based application. This integration is expected to make it easier for users, such as farmers, traders, and processing industry players, to quickly, accurately, and in real time identify chili quality. This study also uses a chili image dataset with diverse lighting and background variations, so the resulting model is expected to be more robust to real-world field conditions. Thus, this research not only offers a technology-based solution that is more modern than conventional edge detection methods but also makes a tangible contribution to the implementation of smart agriculture in Indonesia.

Based on a review of previous studies, a research gap can be identified in studies on chili quality detection based on digital images. Most previous research has focused on feature extraction using edge detection and contour extraction techniques, such as the Canny and Sobel methods [12][13], which are not yet capable of automatically classifying chili quality. Other studies have applied Convolutional Neural Networks (CNNs) and deep learning approaches to detect chili maturity or presence [17][18], however, these studies are still limited to specific chili types, object detection tasks, or do not classify chili quality into more detailed quality categories. In addition, most of these studies have not integrated CNN models into web-based applications that can be directly and real-time accessed by users in the field.

Therefore, this study addresses the identified research gap by developing a CNN model specifically designed to classify chili quality into three categories, namely good, pest-infested, and unknown, and by integrating the model into a web-based application. Accordingly, this research has a clear position as an extension of previous studies in the context of chili quality detection based on digital images.

The novelty of this study lies in the application of a Convolutional Neural Network (CNN) for classifying chili quality into three categories, namely good, pest-infested, and unknown, which is directly integrated into a web-based application to enable real-time usage. In contrast to previous studies that generally focused only on edge detection, chili type identification, or maturity level assessment without application system implementation, this study emphasizes quality classification and practical field implementation.

The purpose of this study is to develop a CNN-based system capable of automatically classifying chili quality into specific categories (good, pest-infested, or low-quality) to support a more objective, consistent, and efficient quality determination process within the agricultural supply chain.

## II.METHOD

The following are the research method steps that have been carried out, which are presented in the diagram in Figure 1:

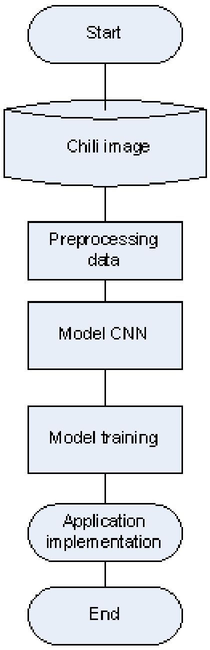


Figure 1. Research Methodology

1. Dataset
   In this study, the dataset used consisted of chili images obtained through observations captured personally. There were 405 chili images representing 11 varietal categories, labeled as ‘good,’ ‘pest-infested,’ or ‘unknown,’ with varying quantities for each image.

Table 1. Sample Chili Image Data

| | | |
|---|---|---|
| Green Bird’s Eye Chili 1 |  | |
| Green Bird’s Eye Chili 2 |  | |
| Green Bird’s Eye Chili 3 |  | |
| Bird’s Eye Chili 1 |  | |
| Bird’s Eye Chili 2 |  | |
| Bird’s Eye Chili 3 |  | |
| Curly Red Chili 1 |  | |
| Curly Red Chili 2 |  | |
| Curly Red Chili 3 |  | |
| Large Green Chili 1 |  | |
| Large Green Chili 2 |  | |
| Large Green Chili 3 |  | |

Table 1 presents the list of chili image files used in this study. These images include various types of chili, such as green bird’s eye chili, curly red chili, and large green chili. Each image has a different file size, depending on the visual complexity and the resolution of the image produced by the phone camera.

2. Preprocessing data

Before the images were used for model training, several preprocessing steps were carried out to prepare the data to meet the requirements of the deep learning model. Before being used in model training, several preprocessing steps were applied to the images. First, all images were resized to 128×128 pixels to ensure consistency in image size used in the model. Next, the pixel values of the images were normalized to the range [0, 1] by dividing each pixel value by 255, aiming to accelerate the model training process. In addition, to increase the variation of training data and prevent overfitting, data augmentation was performed using techniques such as rotating images within a certain angle range, horizontal flipping to add variation to the image appearance, as well as zooming and shearing to enhance the visual complexity of the images.

3. Model CNN

To classify chili quality, a Convolutional Neural Network (CNN) model was used, consisting of several convolutional layers to extract important features from the images. The convolutional layers function to detect basic patterns in the images, such as lines, textures, and shapes, using the ReLU (Rectified Linear Unit) activation function to introduce non-linearity into the model. After the convolutional layers, pooling layers were applied using the max-pooling technique to reduce image dimensions and accelerate computation while retaining important features detected in the convolutional layers. Subsequently, the images processed by the convolutional and pooling layers were flattened and passed through fully connected (dense) layers for classification. The output of this layer is the probability that the image belongs to one of the categories: ‘Good,’ ‘Pest-Infested,’ or ‘Unknown’. In the output layer, the Softmax activation function was used to provide probabilities for each class. For model optimization during training, the Adam optimizer was employed, while Categorical Cross-Entropy Loss was used to measure the difference between the model’s predictions and the actual class labels, as shown in the CNN architecture below :

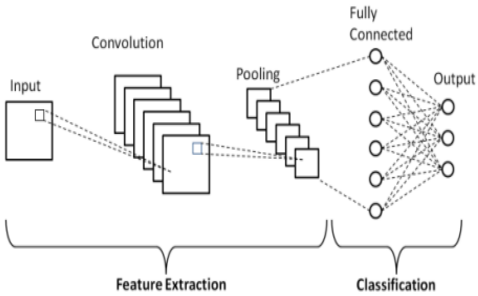


Figure 2. CNN Architecture

In the convolutional layer of a Convolutional Neural Network (CNN), the main process involves applying filters or kernels to the input image to extract important features such as lines, textures, and shapes. These filters operate by sliding across the image (a process called convolution) and performing a dot product operation between the pixel values of the image and the pixel values of the filter. The result of this multiplication is summed to produce a single output value at a specific position. This process is performed on every part of the image, generating features that can help in recognizing specific objects or patterns. These filters are highly effective at detecting local patterns in the image, such as lines, corners, and textures. The basic formula for calculating the output size of a convolutional layer depends on the input image size, filter size, and other parameters such as stride and padding used in the convolution operation. The basic formula for calculating the output size of a convolutional layer is :

$$Output\ Size = \frac{Input\ - Filter\ + 2\ X\ Padding}{\text{Stride}} + 1$$

4. Model *training*

In the model training stage, the constructed CNN model is trained using the preprocessed training data. The training process is carried out to teach the model how to recognize and classify chili images based on the features extracted from those images.

First, the training set was used to train the model. The CNN model was optimized using the Adam optimizer, which is one of the popular optimization algorithms that can accelerate convergence and reduce prediction errors. In addition, Categorical Cross-Entropy Loss was used as the loss function to calculate the difference between the model’s prediction results and the correct labels.

During training, the model goes through several epochs (training iterations), where each epoch consists of processing one batch of data at a time. To prevent the model from overfitting, a validation set is used to monitor the model’s performance during training and adjust its parameters so that it does not over-adapt solely to the training data.

In addition, early stopping can be applied to halt training early if the model has already achieved optimal performance on the validation set, in order to avoid prolonged training without significant performance improvement.

5. *Application implementation*

After the CNN model has been trained and saved in the .h5 format, the next step is to implement the model into a web-based application that allows users to upload chili images and automatically obtain predictions of their quality

(whether the chili is good, pest-infested, or unknown).
For this implementation, Flask, a lightweight web framework for Python, was used, which allows for rapid web application development. The following are the steps for implementing the application: Flask setup begins with installing the Flask framework in the development environment. To start, first install Flask using pip (Python Package Index) with the following command :

```
[ ]  pip install flask
```

Figure 3. Flask pip Installation

After Flask is installed, the next step is to create the main web application file, named app.py. In this file, we will import Flask and create an application object. Below is an example of the basic Flask application setup :

```
[ ] from flask import Flask, render_template, request
    app = Flask(__name__)

    @app.route('/')
    def home():
        return render_template('index.html')

    if __name__ == '__main__':
        app.run(debug=True)
```

Figure 4. Flask Router

In the example above, the app object is an instance of the Flask class that will handle all application URL routes. The @app.route('/') method sets the URL for the application's main page. The home() function is used to render the index.html template file, which will be displayed when the user accesses the main page. After setting up the folder structure and Flask configuration, the next step is to create the main application file named app.py. Below is an example of creating the app.py file for a Flask-based application that integrates the CNN model and allows users to upload chili images and obtain prediction results.

```
from flask import Flask, render_template, request
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
from werkzeug.utils import secure_filename

app = Flask(__name__)
UPLOAD_FOLDER = 'static/uploads'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

model = load_model('model/cabai_model.h5')
labels = ['Bagus', 'Hama', 'Tidak Diketahui']
```

Figure 5. app.py Script

## III.RESULT AND DISCUSSION

After obtaining the dataset, the next step is preprocessing before applying it to the deep learning model. The images are resized to 128×128 pixels and normalized within a value range of 0 to 1. The results of this preprocessing are then used to train the model that will be tested.

The proposed Convolutional Neural Network (CNN) model for detecting chili quality levels was evaluated using a dataset consisting of 11 chili images of various types, including green and red chilies. Model Convolutional Neural Network (CNN) yang diusulkan untuk mendeteksi tingkat kualitas cabai dievaluasi menggunakan dataset yang terdiri dari 11 gambar cabai berbagai jenis, termasuk cabai hijau dan cabai merah. The images were resized to 128×128 pixels, and the pixel values were normalized within the range [0, 1] to accelerate the training process. Data augmentation techniques, such as image rotation, horizontal flipping, zooming, and shearing, were applied to create variation and prevent overfitting during model training. In general, these stages can be described as follows :

1. *Loading* dan *Preprocessing* Data
   In the initial stage, chili images located in the specified folder were loaded and processed using PIL and NumPy. The images were resized to 128×128 pixels and then normalized so that the pixel values fall within the range [0, 1].
2. Model CNN
   The CNN model was built with several convolutional and max-pooling layers for feature extraction, followed by fully connected layers for classification. The configuration details of each layer along with the number of parameters can be seen in Figure 6.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d (Conv2D) | (None, 126, 126, 32) | 896 |
| max_pooling2d (MaxPooling2D) | (None, 63, 63, 32) | 0 |
| conv2d_1 (Conv2D) | (None, 61, 61, 64) | 18,496 |
| max_pooling2d_1 (MaxPooling2D) | (None, 30, 30, 64) | 0 |
| conv2d_2 (Conv2D) | (None, 28, 28, 128) | 73,856 |
| max_pooling2d_2 (MaxPooling2D) | (None, 14, 14, 128) | 0 |
| flatten (Flatten) | (None, 25088) | 0 |
| dense (Dense) | (None, 128) | 3,211,392 |
| dense_1 (Dense) | (None, 2) | 258 |

**Total params:** 3,304,898 (12.61 MB)
**Trainable params:** 3,304,898 (12.61 MB)
**Non-trainable params:** 0 (0.00 B)

Figure 6. Model Summary

As shown in Figure 3, the CNN model consists of three convolutional (Conv2D) layers, each followed by a max-pooling layer to reduce feature dimensions. After the feature extraction process, the results are flattened through a Flatten layer so they can be passed to the fully connected (Dense) layers. The first Dense layer serves as a hidden layer with 128 neurons, while the final Dense layer is the output layer with 2 neurons representing two classes (e.g., good-quality chili and poor-quality chili).

The total number of model parameters is 3,304,898, all of which are trainable, so the weights in each layer will be updated during the training process. This relatively large number of parameters indicates the model's complexity in extracting patterns from chili images.

3. Pelatihan Model
   The model was trained using the training and validation data, with a specified number of epochs and batch size.

```
Epoch 1/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 3s 3s/step - accuracy: 1.0000 - loss: 0.6822 - val_accuracy: 1.0000 - val_loss: 0.0099
Epoch 2/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 318ms/step - accuracy: 1.0000 - loss: 0.0092 - val_accuracy: 1.0000 - val_loss: 7.3909e-06
Epoch 3/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 382ms/step - accuracy: 1.0000 - loss: 6.7502e-06 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 4/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 815ms/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 5/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 579ms/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 6/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 721ms/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 7/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 647ms/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 8/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 353ms/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 9/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 604ms/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 10/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 343ms/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
```

Figure 7. Epoch Process

Figure 7 shows the CNN model training process over 10 epochs. The training accuracy and validation accuracy consistently reached 100% from the first epoch, with loss and validation loss values continuously decreasing toward zero. This indicates that the model successfully learned the patterns in the data very well.

4. Model Evaluation
   After training was completed, the model was evaluated using the test data, and metrics such as accuracy, precision, recall, and F1-score were calculated.

```
Training set shape: (8, 128, 128, 3), Labels shape: (8,)
Validation set shape: (2, 128, 128, 3), Labels shape: (2,)
Testing set shape: (2, 128, 128, 3), Labels shape: (2,)
Evaluating model on test data...
Test Loss: 0.0000
Test Accuracy: 1.0000
Making predictions on test data...
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 128ms/step
Calculating evaluation metrics...
Precision: 1.0000
Recall: 1.0000
F1-score: 1.0000
```

Figure 8. Model Evaluation

Figure 8 shows the evaluation results of the CNN model using the test data. The Test Loss value was recorded as 0, while the Test Accuracy reached 100%. In addition, all other evaluation metrics—precision, recall, and F1-score—also showed perfect scores (1.0000). This means the model was able to classify all test images correctly without errors.

5. Web Application Implementation with Flask
   After the model was trained, the web application was built using Flask, allowing users to upload chili images and obtain prediction results.

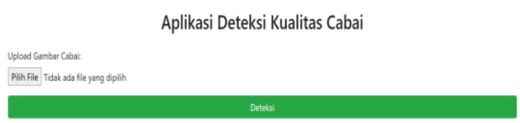


Figure 9. Chili Quality Detection Application Interface

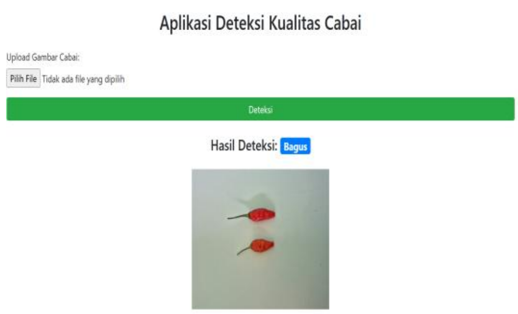


Figure 10. Good Detection Result

In Figure 10, the Convolutional Neural Network (CNN) model successfully detected chili with an optimal quality level. The detection process was carried out using digital chili images that had undergone several stages, including convolution, pooling, and classification. The detection results indicate that the chili has good quality, as evidenced by the identified visual characteristics such as color, shape, and texture that match the ‘Good’ category. The CNN model provides accurate predictions thanks to training with a dataset that includes various types of chilies with different lighting and background variations.

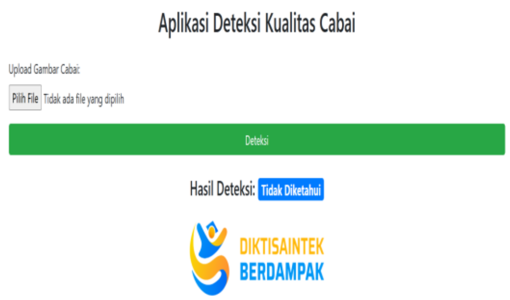


Figure 11. Unknown Detection Result

In Figure 11, the Convolutional Neural Network (CNN) model failed to provide a clear classification of the chili’s quality. This could be due to several factors, such as suboptimal lighting, overly complex backgrounds, or variations in chili shape and texture not covered in the training data. When the model cannot identify with certainty, it returns the status ‘Unknown.’ These results highlight the importance of improving the dataset and enhancing image preprocessing to increase detection capability under more varied conditions.

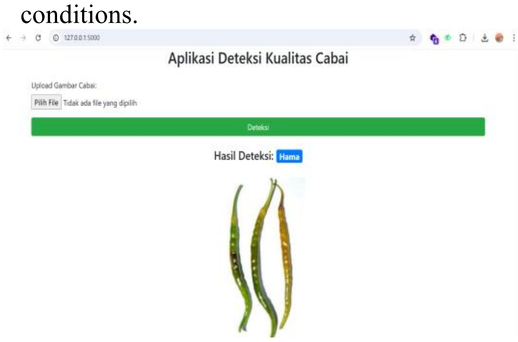


Figure 12. Pest Detection Result

In Figure 12, the Convolutional Neural Network (CNN) model successfully detected a pest-infested chili based on visual characteristics such as color changes, abnormal texture, or surface damage. This detection was performed after the chili image passed through convolution and pooling stages to extract important features. The model classified the chili as ‘Pest-Infested’ by matching the observed patterns with the training dataset that included images of damaged or pest-infested chilies. These results demonstrate the effectiveness of CNN in detecting quality issues in chilies, which is crucial for maintaining agricultural product quality.

The results of this study indicate that the developed Convolutional Neural Network (CNN) model achieved an accuracy of 100% on the test data in classifying chili quality based on digital images. This finding is consistent with the study conducted by Andayani et al. [17], who applied a CNN model using DenseNet169 transfer learning for chili maturity detection and reported high accuracy. The similarity of these results demonstrates that CNNs are highly effective in extracting complex visual features from chili images.

However, the approach employed in this study differs from the methods used by Caya et al. [12] and Rahayu et al. [13], which relied on edge detection techniques such as Canny and Sobel. These methods primarily focus on extracting object contours and edges, resulting in limitations in automatically classifying chili quality. In contrast, CNNs are capable of automatic and comprehensive feature extraction without requiring manual feature selection, leading to more optimal classification performance.

Furthermore, the study by Madupalli et al. [18], which utilized YOLOv5 for red chili detection, mainly focused on object detection rather than on quality classification based on physical condition and pest infestation. This research complements previous work by providing chili quality classification into three categories, namely good, pest-infested, and unknown, thereby offering more comprehensive information for users.

Thus, the findings of this study not only reinforce previous research on the effectiveness of CNNs in agricultural image processing, but also contribute novel value through the integration of a CNN-based chili quality classification system into a web-based application that can be used in real-time under field conditions.

## IV.CONCLUSION

In this study, a chili quality detection system based on a Convolutional Neural Network (CNN) was developed to accurately classify chili quality levels using digital images. This system addresses the limitations of traditional methods that rely on manual visual inspection, which is often subjective and inconsistent. The constructed CNN model is capable of recognizing various types of chilies under different lighting and background conditions, providing more reliable and faster results. The implemented web-based application allows users, such as farmers and traders, to upload chili images and obtain real-time quality predictions. This research makes a significant contribution to the development of smart agriculture technology in Indonesia by providing an automated solution that simplifies the process of chili quality identification. For future research, expanding the dataset with a greater variety of chili types and applying the model in more diverse field conditions could be important steps to improve the accuracy and reliability of this detection system. Thus, this study directly fills the identified research gap by integrating CNN-based chili quality classification with a real-time web-based application, which has not been addressed in previous studies.

## ACKNOWLEDGMENT

The authors would like to express their gratitude to Universitas Bina Sarana Informatika for academic and administrative support, to colleagues and supervisors for their valuable suggestions and input, to the J-TIT journal management team, and to family and friends for their moral support and prayers throughout the course of this research.

## REFERENCE

[1] Sudirman, *PENGENALAN DESKRIPSI VARIETAS CABAI MERAH BESAR*. PT. Nasya Expanding Management, 2022.

[2] W. Zam, I. Ilyas, and S. Syatrawati, "Application of post-harvest technology to increase the selling value of chilies in Tanatoraja," *J. Dedik. Masy.*, vol. 2, no. 2, pp. 92–100, 2019.

[3] L. Soesanto, *Kompendium Penyakit-Penyakit Cabai*. Andi, 2024.

[4] A. D. Zahara T, N. S. Wisnujati, and E. Siswati, "ANALISIS PRODUKSI DAN PRODUKTIVITAS CABAI RAWIT (Capsicum frutescens L) di INDONESIA," *J. Ilm. Sosio Agribis*, vol. 21, no. 1, pp. 18–29, 2021, doi: 10.30742/jisa21120211345.

[5] D. L. C. Sibarani, "Red Chili Leaf Disease Identification via GLCM and SVM Analysis," 2023, doi: 10.1109/ICoSNIKOM60230.2023.10364511.

[6] Y. Pei and Yingchao Zhang, "A Study on the Integrated Development of Artificial Intelligence and Tourism from the Perspective of Smart Tourism," *Yingchao Zhang*, vol. 1852, no. 3, 2021, doi: https://doi.org/10.1088/1742-6596/1852/3/032016.

[7] M. R. Almadani, P. Prajoko, and D. Indriyana, "Klasifikasi Kesehatan Tanaman Padi Menggunakan Algoritma Convolutional Neural Network," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 5, 2024, doi: https://doi.org/10.36040/jati.v8i5.10844.

[8] S. Mulyana, M. AS, A. Warjaya, I. Muthmainnah, Said Iskandar Al Idrus, and Zulfahmi Indra, "Identifikasi Penyakit Tanaman Berdasarkan Citra Daun Berbasis Web dengan Pendekatan Algoritma Convolutional Neural Network," *SKANIKA Sist. Komput. dan Tek. Inform.*, vol. 8, no. 2, 2025, doi: https://doi.org/10.36080/skanika.v8i2.3573.

[9] R. Kumari, A. Kumar, H. L. Mandoria, B. K. Pandey, and Subodh Prasad, "A Review on Comparative Study of Different Edge Detection Techniques," *Int. Res. J. Eng. Technol.*, vol. 7, no. 10, 2020.

[10] O. T. Tan, Sean Huey ; Lam, Chee Kiang ; Kamarudin, Kamarulzaman ; Ismail, Abdul Halim ; Rahim, Norasmadi Abdul ; Azmi, Muhamad Safwan Muhamad ; Yahya, Wan Mohd Nooriman Wan ; Sneah, Goh Kheng ; Seng, Moey Lip ; Hai, Teoh Phaik ; Lye, "Vision-Based Edge Detection System for Fruit Recognition," *J. Phys. Conf. Ser.*, vol. 2107, no. 1, 2021, doi: 10.1088/1742-6596/2107/1/012066.

[11] T. S. A. M. K. C. V. E. D. P. K. V. M. A. S. P. E. S. M. J. M. Lorenzo, "Edge Detection Aided Geometrical Shape Analysis of Indian Gooseberry (Phyllanthus emblica) for Freshness Classification," *Food Anal. Methods*, vol. 15, pp. 1490–1507, 2022, doi: https://doi.org/10.1007/s12161-021-02206-x.

[12] C. D. C. S. M. C. M. V. Caya, "Identification of Philippine Chili Peppers using Canny Edge Detection and Feed– Forward Artificial Neural Network," 2023, doi: 10.1109/HNICEM57413.2022.10109560.

[13] S. Rahayu, R. Ridwansyah, and J. J. Purnama, "Analisis Pengaruh Threshold Pada Metode Canny Dan Sobel Dalam Deteksi Tepi Citra Cabai," *J. Teknoinfo*, vol. 19, no. 2, 2025, doi: https://doi.org/10.33365/teknoinfo.v19i2.162.

[14] N. C. M. K. T. G. P. P. J. C. W. Yimyam, "Sorting Red and Green Chilies by Digital Image Processing," 2023, doi: 10.1109/ICCI57424.2023.10112254.

[15] B. Falakhi, E. F. Achmal, M. Rizaldi, R. R. R. Athallah, and N. Yudistira, "Perbandingan Model AlexNet dan ResNet dalam Klasifikasi Citra Bunga Memanfaatkan Transfer Learning," *J. Ilmu Komput. dan Agri-Informatika*, vol. 9, no. 1, pp. 70–78, 2022, doi: 10.29244/jika.9.1.70-78.

[16] A. P. B. Salsabila, C. Rozikin, and R. I. Adam, "Klasifikasi Motif Batik Karawang Berbasis Citra Digital dengan Principal Component Analysis dan K-Nearest Neighbor," *J. Sist. dan Teknol. Inf.*, vol. 11, no. 1, p. 20, 2023, doi: 10.26418/justin.v11i1.46936.

[17] A. B. K. M. R. I. R. A. T. A. M. M. Y. A. D. D. D. Andayani, "Detection of Katokkon Chili Maturity using Convolutional Neural Network with Transfer Learning Model DenseNet169," 2024, doi: 10.1109/IWAIIP58158.2023.10462730.

[18] P. P. V. S. R. K. V. S. Y. S. K. B. S. S. M. Madupalli, "Deep Learning Model YOLOv5 for Red Chilies Detection from Chilly Crop Images," 2023, doi: 10.1109/I2CT57861.2023.10126327.

[19] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles," 2023, [Online]. Available: http://arxiv.org/abs/2112.11798.