Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 24, No. 3, July 2025, pp. 465∼478
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 10/C/C3/DT.05.00/2025
DOI: 10.30812/matrik.v24i3.4464

A Novel CNN-Based Approach for Classification of Tomato Plant Diseases

Miftahus Sholihin[1], Mohd Farhan Md Fudzee[2], Lilik Anifah[3]
[1]Universitas Islam Lamongan, Lamongan, Indonesia
[2]University Tun Hussein Onn Malaysia, Johor, Malaysia
[3]Universitas Negeri Surabaya, Surabaya, Indonesia

Article Info

Article history:

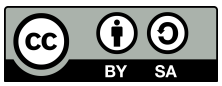
Received September 11, 2024
Revised June 18, 2025
Accepted June 24, 2025

Keywords:

Accuracy;
Convolution Neural Network;
Model;
Performance;
Tomato.

ABSTRACT

Tomatoes are one of the most widely cultivated and consumed crops globally, but they are highly susceptible to various diseases that can significantly reduce yield and quality. Early detection of these diseases is crucial for effective management and prevention. **The objective of this study** is to develop an accurate early detection system for tomato diseases using deep learning to support effective crop management. **The research method** employed is a modified Convolutional Neural Network trained on the PlantVillage dataset, which consists of 21,000 images across 10 disease classes. The study evaluates three training scenarios using different epoch values (25, 50, and 75) to optimize model performance. Data preprocessing included image resizing and augmentation, followed by Convolutional Neural Network training and validation. **The study's results** showed that increasing epochs improved the model's accuracy: 98.18% at 25 epochs, 98.53% at 50 epochs, and 99.19% at 75 epochs. Precision, recall, and F1-score also increased, from 90.95% at 25 epochs to 95.80% at 75 epochs, indicating enhanced model reliability. However, longer training times were required as the epoch count increased. **This research concludes that a modified** Convolutional Neural Network can accurately classify tomato diseases, providing a reliable and practical tool for early disease detection. The proposed system has the potential to be integrated into mobile applications for real-time use in the field. It contributes to sustainable agriculture by enabling timely disease intervention and improving crop productivity.



Corresponding Author:

Miftahus Sholihin, +6283832347831,
Faculty of Engineering, Informatics Engineering,
Universitas Islam Lamongan, Lamongan, Indonesia,
Email: miftahus.sholihin@unisla.ac.id.

How to Cite:
M. Sholihin, M. F. Bin Md. Fudzee, and L. Anifah, "A Novel CNN-Based Approach for Classification of Tomato PlantDiseases, "A Novel CNN-Based Approach for Classification of Tomato Plant Diseases", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 24, No. 3, pp. 163-176, July, 2025.

1. INTRODUCTION

Tomatoes (Solanum lycopersicum) are among the most widely cultivated horticultural crops globally due to their high nutritional and economic value. However, disease outbreaks remain a major constraint in maximizing tomato yield and quality [1, 2]. This pharmacological effect can protect a person from diseases such as hepatitis, hypertension, and gingival bleeding. In addition, tomatoes are rich in vitamins A and C and antioxidants such as lycopene. China became the country with the highest tomato production in the world in 2022, with a production of 68 million tons. One of the factors that can affect tomato production is disease. Diseases that attack tomato plants significantly impact the quality of their production, and early detection is needed to identify and reduce the risk of damage. In general, early detection is usually carried out by agricultural or botanical experts who require special expertise and take a long time, so it is not efficient to be applied widely [3–5]. With the high potential for errors in manual early detection, the implementation of automatic early detection is very crucial to speed up handling and minimize errors in the detection process [6]. With advances in technology in deep learning, especially in image processing, there is an opportunity to develop an automated system to identify plant diseases with high accuracy [7].

In recent years, there have been many studies using deep learning to classify types of diseases in tomato plants [8–10]. The use of deep learning methods has been proven to provide better results when compared to traditional machine learning methods [11, 12]. Research conducted by Karthik [13] using a Residual Convolutional Neural Network (CNN) to classify three types of diseases in tomato plants with an accuracy of 98%. Although the results of this study show high accuracy, by only focusing on three types of diseases, this model is less applicable to more complex types of diseases. Argawal [14] tries to overcome the scope of the number of diseases used, where in his study, he used ten types of diseases, and CNN was used as the classification method. This study provides an accuracy of 91.2%. Trivedi [15] also used ten types of diseases and the CNN method to classify tomato plant disease types with an accuracy of 98.42%. Anandhakrishnan [16] used the deep convolutional neural network method to classify tomato plant diseases with an accuracy of 98.40%. These results indicate that CNN is adequate for classifying tomato plant disease types. Islam [17] used CNN to classify tomato plant disease types with an accuracy of 99% for the training process, 97.5% for validation, and 98% for the testing process. Abbas [18] used the conditional generative adversarial network (C-GAN) method to detect diseases in tomato plants. This study showed an accuracy of 99.51% for five classes, 98.65% for seven classes, and 97.11% for ten classes. However, the complexity of the C-GAN architecture and high computational requirements pose challenges in its widespread application. Chen [19] used the Binary Wavelet Transform method combined with Retinex (BWTR) to remove noise. Furthermore, the Artificial Bee Colony (ABCK) algorithm separated tomato leaves from the background. The final process is classification using the Both-channel Residual Attention Network (B-ARNet) model. This study achieved an accuracy of 89%.

Furthermore, Sakkarvarthi [20] used the CNN method for tomato plant disease classification. During training, the model created provided an accuracy of 98%, but during testing on new data, the model created only provided an accuracy of 88.17%. This indicates a problem in the classification process, where the model created cannot generalize to new data well. Meanwhile, Badiger [21] created a model known as GJ-GSO + DbneAlexNet, where this algorithm is a modified algorithm from a neural network and an optimization algorithm called Gradient Jaya-Golden Search Optimization (GJ-GSO). This study also uses U-Net for the image segmentation process. This study provides an accuracy of 92.4%. Baser [22] also modified the CNN architecture to increase the accuracy in classifying tomato plant diseases. This study provides an accuracy of 98.19% with ten classes. Meanwhile, Yang [23] used the LSGNet (lightweight convolutional neural network) method for tomato plant disease classification with an accuracy of 95.54%.

The novelty of this study lies in its systematic evaluation of epoch variations (25, 50, and 75) on a modified CNN architecture, which is rarely explored in previous research. In addition, this study focuses on achieving high accuracy, addressing generalization issues, and reducing computational cost, which are common limitations in prior works. This study addresses the identified gaps by developing a modified CNN architecture that improves classification accuracy and generalization. Unlike existing models that suffer from overfitting or high computational cost, the proposed architecture employs four convolutional layers, two dense layers, and a dropout regularization strategy (50%) to enhance learning without overfitting. In addition, the study systematically investigates epoch values (25, 50, and 75) to analyze their impact on model performance and training time. This hyperparameter exploration revealed that 75 epochs yielded the best results, achieving 99.16% accuracy, precision, recall, and F1-score, all at 95.8%, outperforming prior methods using the same dataset. These findings demonstrate that the proposed CNN is accurate and practical, balancing performance and computational cost. The model is especially suitable for real-world agricultural deployment, where computational resources may be limited but accurate early disease detection is essential, Despite the promising results of previous CNN-based models, many of these studies lack systematic exploration of training hyperparameters, especially epoch variation, and often struggle with overfitting or high computational complexity, limiting real-world applicability.

The implications of this research are significant for both the agricultural sector and the broader field of deep learning. For farmers, developing an accurate and efficient automated disease detection system can lead to early identification of plant diseases,

enabling timely intervention and reducing crop losses. This, in turn, can improve tomato production quality and yield, contributing to food security and economic benefits. For the scientific community, this research contributes to advancing deep learning techniques in agriculture, particularly optimizing CNN architectures for image classification tasks. The findings can serve as a foundation for future studies aiming to develop lightweight, high-performance models for other crop diseases or even other domains requiring image classification. Ultimately, this research bridges the gap between technological innovation and practical agricultural applications, fostering sustainable farming practices.

2. RESEARCH METHOD

This section concisely describes the proposed method for tomato disease classification. The research process began with collecting relevant data, the initial and significant step. After the data was collected, the next stage was data pre-processing, where the collected data was cleaned, processed, and prepared so that it was ready to be used in the model. The next stage was model building. Appropriate methods and techniques were applied to build a classification model at this stage. Based on the processed data, this model was designed to identify and classify various types of diseases in tomato plants. After the model was built, the last stage was model testing. The model was tested in this stage to assess its accuracy and effectiveness in classifying tomato plant diseases. This testing was essential to ensure the model could provide accurate and reliable results in actual practice. This testing is essential to ensure the model can provide accurate and reliable results in actual practice. Figure 1 presents a general flowchart of the research conducted.

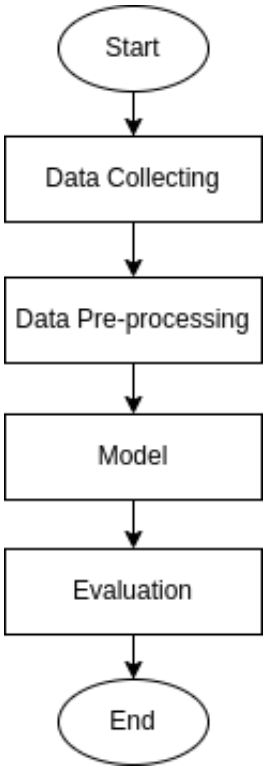


Figure 1. Research framework

2.1. Data Collecting

The dataset used in this research comes from the Plant Village dataset, available online and can be downloaded at https://data.mendeley.com/datasets/tywbtsjrjv/1. This dataset contains 38 types of plant diseases. However, this research only focused on types of tomato plant diseases. There were ten types of tomato diseases in this study. Data for each disease consisted of 1000 images in Table 1. So, the total data used in this research is 10,000. Figure 2 is an example of an image of tomato disease.

Table 1. Classes of the Tomato Disease Dataset

Class name	Number of images
Tomato with bacterial spot	1000
Tomato with early blight	1000
Healthy tomato	1000
Tomato with late blight	1000
Tomato with leaf mold	1000
Tomato with septoria leaf spot	1000
Tomato with two-spotted spider mite	1000
Tomato with target spot	1000
Tomato with mosaic virus	1000
Tomato with yellow leaf curl virus	1000

Figure 2. Sample of a tomato leaf

2.2. Data Preprocessing

The data pre-processing stage is crucial for ensuring that the deep learning model constructed has a firm basis and can be generalized effectively. Data augmentation follows the gathering and cleaning of the tomato leaf image dataset. It is a method to raise dataset variability without explicitly adding more data. This is quite crucial in the framework of deep learning, where the variety of the data may significantly influence the capacity of the model to identify pertinent patterns and features [24].

This work used image rotation, horizontal and vertical flipping, zooming, and brightness and contrast changes among several augmentation approaches. These methods enhance the data's variance and replicate several scenarios that tomato leaf photos in the field could encounter: diverse viewing angles, shifting lighting, and varying leaf sizes. This augmentation should help the deep learning model be more robust in handling natural fluctuations not observed in the original training data.

Data augmentation also reduces the likelihood of overfitting, where the model becomes too "familiar" with the training data to lose its capacity to identify patterns in fresh data. Exposing the model to new versions of the same data helps it learn more general and applicable features rather than merely the particular aspects of the original dataset. Training the CNN model then makes use of the outcomes of this augmentation. The added data is supposed to increase the generalization and accuracy capacity of the model in tomato plant disease classification.

2.3. Model Convolutional Neural Network (CNN)

The model developed in this study to classify tomato plant diseases is a modified CNN that deviates from the structure of a standard CNN to enhance performance in classifying tomato plant diseases. The basic CNN architecture generally consists of a few convolutional layers, pooling, and dense layers. In contrast, the proposed architecture introduces a deeper and more structured design, comprising four convolutional layers with increasing filter sizes of 32, 64, 64, and 128, respectively. These layers are responsible for progressively extracting more complex and abstract features from the input images of tomato leaves, as illustrated in Figure 3.

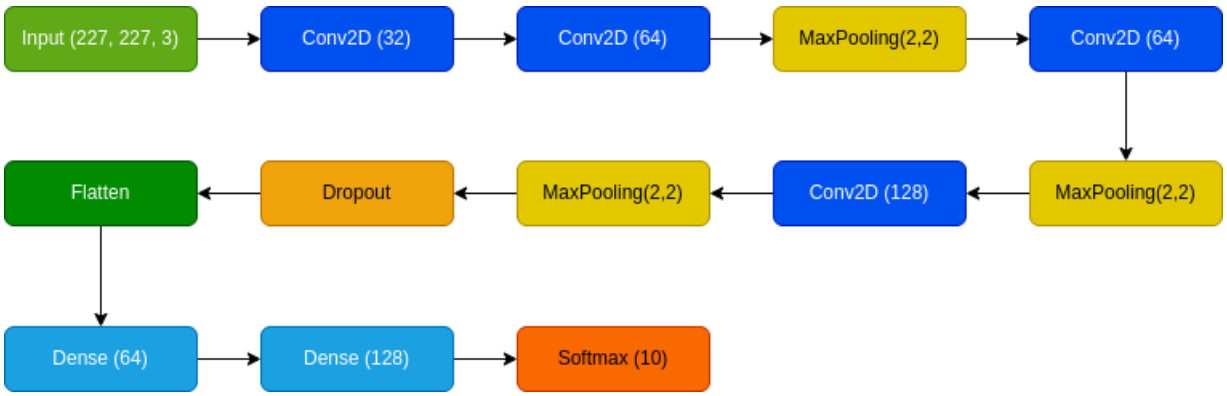


Figure 3. Proposed method

Three MaxPooling layers with a pool size of 2×2 are strategically placed after the first, third, and fourth convolutional layers to reduce the spatial dimensions and computational complexity while retaining the most salient features. Furthermore, a Dropout layer with a 50% drop rate is applied after the final convolutional block to combat overfitting and improve generalization to unseen data. This component is not present in most basic CNN architectures and is essential for increasing robustness.

Following the feature extraction phase, the network transitions to a classification phase, consisting of a Flatten layer and two fully connected (Dense) layers with 64 and 128 neurons, respectively, each using the ReLU activation function. This configuration allows the model to learn complex nonlinear combinations of the extracted features. Finally, a Softmax output layer with 10 neurons is employed to classify the input into one of the ten tomato disease categories.

These modifications—deeper convolutional layers, regularized architecture with dropout, and enhanced dense connections—were designed to improve feature learning, accuracy, and generalization compared to basic CNN models. The impact of these structural changes is reflected in the experimental results, where the model consistently achieved high accuracy and balanced performance metrics, outperforming several other CNN-based approaches reported in the literature.

2.4. Evaluation Performance

A confusion matrix is used to measure the performance of the model created. A confusion matrix is a matrix that can be used to evaluate the performance of a model. This matrix works by comparing actual values with predicted values. The performance is measured using a confusion matrix, which includes accuracy, precision, recall, and F1-score.

1. Precision measures the proportion of correctly predicted positive instances (true positives, TP) out of all instances predicted as positive (true positives plus false positives, FP). It is calculated using Equation 1. This metric is particularly useful when minimizing false positives is critical, such as in disease detection, where incorrect diagnoses could lead to unnecessary treatments.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

2. Recall, also known as sensitivity, evaluates the proportion of correctly predicted positive instances (true positives, TP) out of all actual positive instances (true positives plus false negatives, FN). It is computed using Equation 2. Recall is crucial in applications where missing positive cases, such as undetected diseases, could have severe consequences. A high recall indicates that the model is effective at identifying most of the relevant instances.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

3. Accuracy is a widely used metric that assesses the overall correctness of the model by calculating the ratio of correctly predicted instances (true positives, TP, and true negatives, TN) to the total number of instances (true positives, false positives, FP, true negatives, and false negatives, FN). It is computed using Equation 3. While accuracy provides a general measure of model performance, it may be less informative in imbalanced datasets where one class significantly outweighs the other.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

4. The F1 score is a harmonic mean of precision and recall, providing a balanced measure of a model's performance, especially when there is an uneven class distribution. It is calculated using Equation 4. The F1 score is particularly useful when both false positives and false negatives need to be minimized, as it combines the strengths of both precision and recall into a single metric. This makes it a robust evaluation tool for classification tasks in fields like agriculture, where both over-detection and under-detection of diseases can have significant implications.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

2.5. Experimental Setup

The model created is run on an Intel Core i7 processor and a 16GB RAM computer with the Ubuntu 20.04.4 LTS operating system. The model is implemented in Python with the TensorFlow library. The input image size is 227×227×3, all in RGB format, during the training process using the following parameters: batch size 32, learning rate 0.001, and Adam optimizer. Meanwhile, the epoch used in this research uses three different values, namely 25, 50, and 75. This research divides the dataset into training, validation, and testing with a percentage ratio of 70:10:20. The test scenarios carried out in this research are shown in Table 2.

The selection of 25, 50, and 75 epochs in the experimental setup was based on exploring how different training durations affect the model's learning ability, accuracy, and generalization. A lower number of epochs, such as 25, was chosen to represent a short training cycle, which is helpful to assess how quickly the model can converge and whether early stopping might be sufficient. It also helps to understand the model's baseline performance with minimal training. An intermediate value of 50 epochs was selected as a standard reference point, commonly used in many deep learning studies, to evaluate whether extending training beyond the minimum improves model performance significantly without introducing overfitting. This helps strike a balance between training time and model performance. The higher value of 75 epochs was selected to observe whether additional training improves accuracy and generalization or leads to diminishing returns and potential overfitting. By comparing performance across these three points, this study aimed to identify the optimal number of epochs for training the modified CNN architecture. The variation in epoch values also enabled analysis of the trade-off between accuracy improvement and computational cost, which is essential for practical deployment in resource-constrained environments.

Table 2. Test Scenario

Scenario	Parameter
Scenario 1	epoch 25, batch size 32, learning rate 0.001, Adam optimizer
Scenario 2	epoch 50, batch size 32, learning rate 0.001, Adam optimizer
Scenario 3	epoch 75, batch size 32, learning rate 0.001, Adam optimizer

3. RESULT AND ANALYSIS

In this section, trials are carried out on the model that has been created. The trials in this research were divided into several scenarios according to the scenarios in subchapter 2.5. The following are the results of trials carried out with several scenarios.

3.1. Scenario 1

Testing carried out in this scenario begins with training the model created. This process aims to provide knowledge to the model developed in recognizing the characteristics of the training data. Apart from that, this process is also used to identify patterns from images that represent certain classes. The model created is based on the information provided during the training process. Figure 4 is an image of the results of the training process of the model created. Based on Figure 4, training accuracy, training loss, and validation loss can be seen. Training accuracy is a metric used to measure how well the model predicts data during training. If a model has high accuracy during the training process, then the model can learn the patterns that exist during training. In this study, the training accuracy value was 92.4%. Meanwhile, validation accuracy shows how well the model can recognize data not used during training. The higher validation accuracy value shows that the model created can recognize new data not used for training. The training loss value shows how big the difference is between the predicted value produced by the model and the actual value. Meanwhile, validation loss is used to measure how big the difference is between the expected value and the actual value in the validation data. After completing the training process, the following process tests the model created.

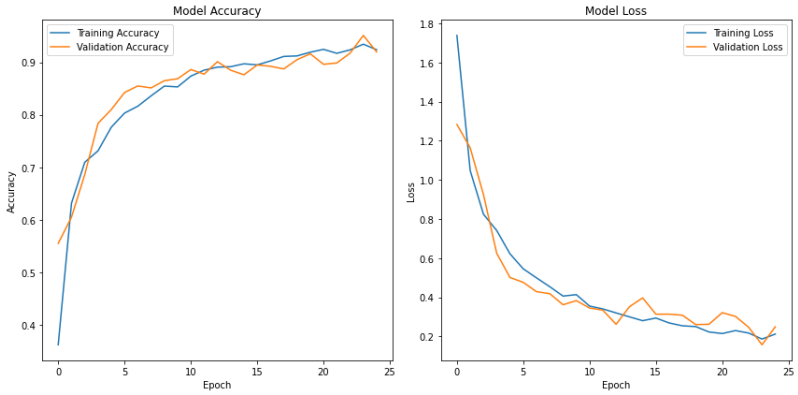


Figure 4. Training, validation accuracy, training loss, and validation loss for scenario 1

The results of this test are shown in the confusion matrix, illustrated in Figure 5. The model evaluation results in this scenario show that the model performance is very good in recognizing mosaic disease types and healthy leaves. For mosaic disease, the model correctly classified 199 out of 200 images, achieving an accuracy rate of 99.5% in this class. Likewise, the model could recognize 199 out of 200 images with the same accuracy of 99.5% for healthy leaves. However, the model performance showed lower results for early blight disease types. of the 200 images tested, the model only succeeded in classifying 160 images correctly, and 40 images could not be classified according to their class, resulting in an accuracy of 80%. Overall, this model shows excellent results, with a total accuracy of 98.19%. The model also has precision, recall, and F1 scores, each of which is 90.95%. These values show that the model is accurate in classification and consistent in detecting all types of diseases included in the dataset.

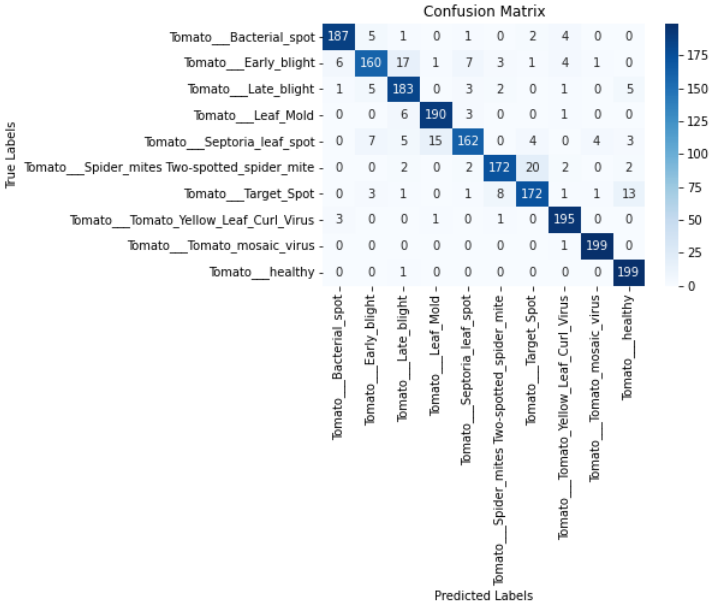


Figure 5. Confusion matrix for scenario 1

3.2. Scenario 2

The parameters used in this scenario are learning rate 0.001, batch size 32, epoch 50, and Adam for the optimizer. Testing carried out in this scenario begins with training the model created. The model's training process results are shown in Figure 6, where the training accuracy value shows how well the model learns from the data to train it. Meanwhile, validation training tests the model's ability to generalize data, which it has never seen before. Training loss shows how far the model predictions are from the

actual values. Meanwhile, validation loss shows the prediction value of the model against validation data. In this study, the training accuracy value was 95.79%. The trial results in this scenario as a confusion matrix, as shown in Figure 7. Based on the confusion matrix, the model can classify the type of mosaic disease entirely. This shows that the model made has a high level of accuracy in recognizing the kind of mosaic disease. In addition, the model created can also identify all types of tomato plants that do not experience disease. Meanwhile, the model made gives the lowest results in recognizing the type of late blight disease. Of the 200 images, 165 can be identified correctly, while 35 cannot be recognized according to the actual class. All of the models that were made show excellent performance. This is indicated by the accuracy value of 98.53%, precision of 92.65%, recall of 92.65%, and F1 score of 92.65%.

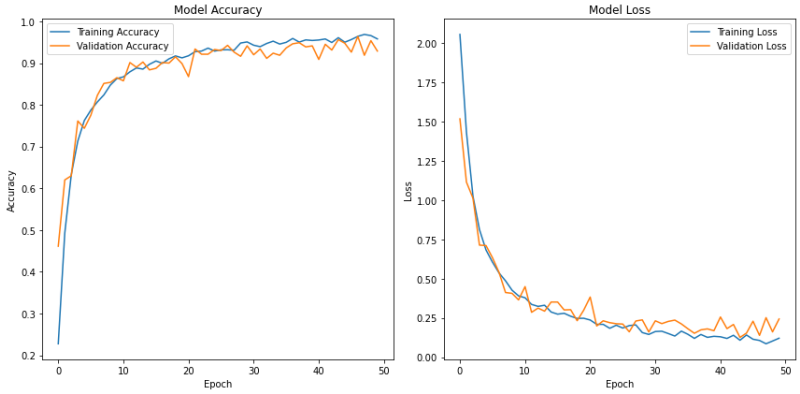


Figure 6. Training, validation accuracy, training loss, and validation loss for scenario 2

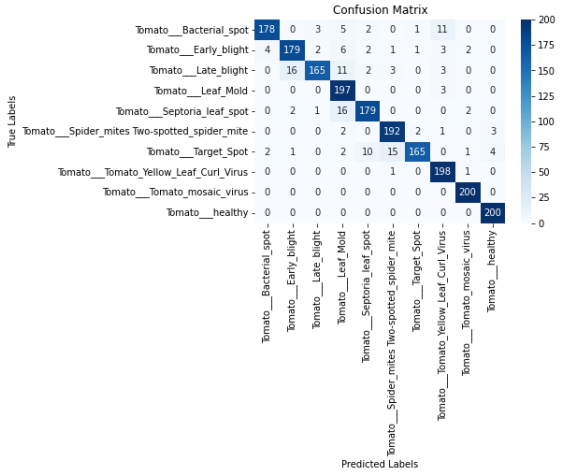


Figure 7. Confusion matrix for scenario 2

3.3. Scenario 3

In this scenario, the tests carried out are similar to those in scenarios 1 and 2. In this scenario, the only difference is the number of epochs used. In this scenario, the epoch used is 75. By using this number of epochs, it is hoped that the model created will experience an increase in predicting data in greater depth. Increasing the number of epochs is expected to improve model performance in the classification process. However, in general, an increase in epochs will be followed by an increase in execution time during the training process. The training process results are shown in Figure 8, which includes the training accuracy, validation accuracy, training loss, and validation loss. The training accuracy value has increased because the model has started to learn from existing data. Training accuracy appears to be stable at epochs 60 to 75. Meanwhile, validation accuracy also increases as the epoch value increases. This shows that the model created not only learns to recognize training data, but also learns data that has never been seen

at all. Meanwhile, the training loss value will decrease. The training loss value decreases drastically at the beginning of the training process. However, this decrease occurs consistently as the epoch value increases. Likewise, the validation loss value also experienced a drastic decline at the start of training. However, this decrease occurs consistently with increasing epoch values. Figure 8 shows that during the training process, the model creates memories of the training data and generalizes new data very well. In this study, the training accuracy value was 97.31%.

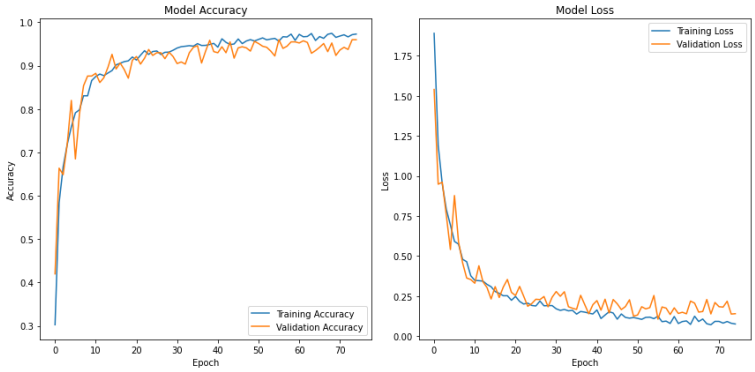


Figure 8. Training, validation accuracy, training loss, and validation loss for scenario 3

Meanwhile, the model's performance in recognizing new data is shown as a confusion matrix in Figure 9. Based on Figure 9, the model created is not yet fully able to correctly recognize the type of tomato plant disease. This late blight disease gives the lowest results compared to other types of disease. Of the 200 images, the model could correctly recognize 180, while the other 20 were incorrectly recognized. Meanwhile, the model gave the highest recognition results for mosaic virus disease types and tomato types unaffected by the disease. The two types of models created could correctly recognize 198 images out of 200 images. In contrast, two images could not be recognized correctly according to their class. Overall, the model created provides very good performance. This is shown by the accuracy value of 99.16%, precision of 95.8%, recall of 95.8%, and F1 score of 95.8%. In this way, the model can recognize new data that it has never seen before very well.

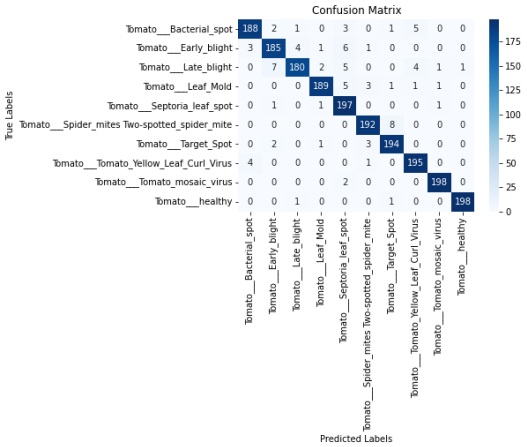


Figure 9. Confusion matrix for scenario 3

3.4. Analysis

Figure 10 is the result of trials for all scenarios (1, 2, and 3). The test uses several different epoch values. The results show that the model created experiences a consistent upward trend in performance as the epoch value increases. The initial epoch value used is 25. With this epoch value, the model created provides an accuracy of 98.19%, and precision, recall, and F1 score values are all 90.95%. These results show that the model's performance is good. However, there is still room for improvement between predicted

positive and negative values, which are measured based on precision and recall values. When the epoch value was set to 50, the model experienced a significant increase in performance. This is shown by increasing accuracy to 98.53% and precision, recall, and F1 scores to 92.65%. This increase shows that the model created can recognize disease types better than using epoch 25. This can happen because the model created requires a longer training time than the epoch 25 value, so the model can learn more details in recognizing a pattern in the image.

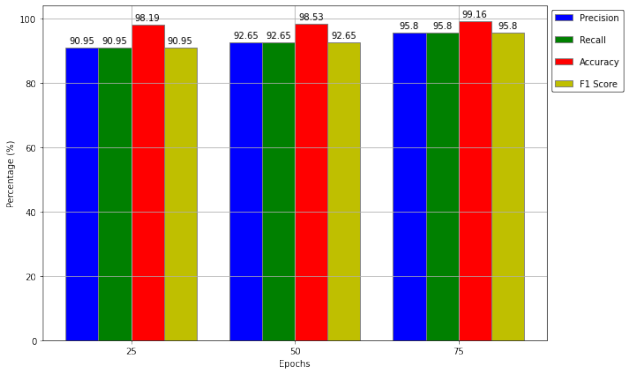


Figure 10. Testing results with epoch values 25, 50, and 75.

The testing process results, as summarized in Table 3, demonstrate the effectiveness of the proposed method compared to previous studies. The proposed method, which utilizes a modified Convolutional Neural Network (CNN) architecture, achieves an accuracy of 99.16% on the PlantVillage dataset for 10 classes of tomato plant diseases during testing. This performance surpasses the testing results of several existing methods, as shown in the table—for instance, Abbas et al. [18] employed DenseNet and Conditional Generative Adversarial Networks (C-GAN) to achieve an accuracy of 97.11% for 10 classes during testing, while Karthik [13] used a Residual CNN to achieve 98% accuracy, but was limited to only three classes. Similarly, Agarwal [14] achieved 91.20% accuracy using a basic CNN for 10 classes during testing, and Sakkarvarthi [20] reported a significant drop in accuracy to 84.78% during testing, highlighting poor generalization.

The proposed method also outperforms other advanced techniques during testing, such as the Attention-Driven Composite Loss Residual Network (ADCLR) used by Islam et al. [17], which achieved 96.60% accuracy, and the CNN model by Trivedi et al. [15], which achieved 98.49% accuracy for 10 classes. The higher accuracy of the proposed method during testing underscores its superior ability to generalize well to unseen data, a critical factor for practical deployment in agricultural settings. This improvement is attributed to the optimized architecture and parameter tuning, particularly the careful adjustment of epoch values, which enhances the model's learning capability and reduces overfitting.

Furthermore, the proposed method's ability to achieve 99.16% accuracy during testing, while maintaining high precision, recall, and F1 scores (all at 95.8%), demonstrates its robustness and reliability. This balance between accuracy and generalization is critical for practical applications, as it ensures that the model performs well on training data and unseen data, which is essential for real-world deployment in agricultural settings. The results also indicate that the proposed method addresses the limitations of previous models, such as poor generalization (as seen in Sakkarvarthi [20]) or limited applicability to a small number of disease classes (as in Karthik [16]).

The proposed method's testing results highlight its superiority over existing approaches, making it a highly effective tool for tomato plant disease classification. Its ability to maintain high accuracy and balanced performance metrics during testing ensures its practicality and reliability for farmers. It enables early and accurate disease detection to minimize crop losses and improve agricultural productivity.

Although the proposed method adopts the Convolutional Neural Network (CNN) as its foundation, it is important to emphasize that this study does not merely reuse standard CNN models previously applied in related works. Instead, a modified CNN architecture is introduced and specifically designed to classify ten classes of tomato leaf diseases using the PlantVillage dataset. The modifications made to the architecture include the integration of four convolutional layers with progressively increasing filter sizes (32, 64, 64, and 128) to enhance feature extraction capabilities. To reduce the spatial dimensions of the feature maps and control overfitting, three MaxPooling layers are inserted at strategic positions within the architecture. A Dropout layer with a dropout rate of 50% is employed to regularize the model and prevent overfitting during training. In the final stage, two fully connected layers are placed before the

output layer to capture complex patterns, followed by a softmax layer for classification. The input size is also adjusted to $227 \times 227 \times 3$ to ensure compatibility with the architecture and dataset.

These modifications enhance the model's ability to extract deep hierarchical features, improve its generalization performance, and reduce overfitting — all of which distinguish it from conventional CNN architectures reported in earlier works, such as those by Agarwal [14], Sakkarvarthi [20], and Trivedi et al. [15]. Therefore, while the backbone remains CNN, this study's architectural design, depth, and regularization strategies are novel and contribute to the improved accuracy and robustness demonstrated in the experimental results.

Table 3. Comparison with Related Studies

Reference	Methods	Dataset	Classes	Accuracy
[18]	DenseNet, C-GAN	PlantVillage	10	97.11%
[13]	Residual CNN	PlantVillage	3	98%
[14]	CNN	PlantVillage	10	91.20%
[20]	CNN	PlantVillage	-	84.78%
[17]	ADCLR	PlantVillage	10	96.60%
[15]	CNN	PlantVillage	10	98.49%
Proposed method	CNN	PlantVillage	10	99.16%

4. CONCLUSION

This study successfully developed a modified Convolutional Neural Network (CNN) classification model for tomato plant disease. The results of trials with various epoch values prove that increasing the number of epochs consistently improves model performance. The model created achieved the highest accuracy of 99.1% at epoch 75, with high precision, recall, and F1 score values of 95.8% each. These results indicate that the model can learn to make predictions with minimal error rates from training data. Compared to previous studies, the model created shows superior performance with higher accuracy, 99.16%. However, increasing epochs also results in a significant increase in training time. Therefore, consideration is needed between increasing accuracy and time efficiency in the practical application of this model. Overall, the model created shows excellent potential in tomato plant disease classification and can effectively support agriculture.

Several aspects can be explored for further development. Although the created model shows high performance, more complex CNN architectures or ensemble techniques can be explored to improve the accuracy and generalization of the model. Optimization techniques such as transfer learning, pruning, or quantization can also be applied to overcome the problem of increasing training time by adding epochs. In addition, testing with more varied datasets is needed to ensure the robustness of the model in real conditions.

5. ACKNOWLEDGEMENTS

Thank you to Universitas Islam Lamongan for providing funding for this research.

6. DECLARATIONS

AUTHOR CONTIBUTION

The first author, Miftahus Sholihin, contributed to data collection and model creation. Mohd Farhan Md Fudzee, the second author, reviewed the model created. Lilik Anifah, the third author, contributed to writing this article.

FUNDING STATEMENT

Thank you to Universitas Islam Lamongan for providing funding for this research.

COMPETING INTEREST

REFERENCES

[1] A. Bhujel, N.-E. Kim, E. Arulmozhi, J. K. Basak, and H.-T. Kim, "A Lightweight Attention-Based Convolutional Neural Networks for Tomato Leaf Disease Classification," *Agriculture*, vol. 12, no. 2, p. 228, Feb. 2022, https://doi.org/10.3390/agriculture12020228.

[2] Q. Wu, Y. Chen, and J. Meng, "DCGAN-Based Data Augmentation for Tomato Leaf Disease Identification," *IEEE Access*, vol. 8, pp. 98 716–98 728, 2020, https://doi.org/10.1109/ACCESS.2020.2997001.

[3] S. Ledbin Vini and P. Rathika, "TrioConvTomatoNet: A robust CNN architecture for fast and accurate tomato leaf disease classification for real time application," *Scientia Horticulturae*, vol. 330, p. 113079, Apr. 2024, https://doi.org/10.1016/j.scienta.2024.113079.

[4] R. Wang, M. Lammers, Y. Tikunov, A. G. Bovy, G. C. Angenent, and R. A. De Maagd, "The rin, nor and Cnr spontaneous mutations inhibit tomato fruit ripening in additive and epistatic manners," *Plant Science*, vol. 294, p. 110436, May 2020, https://doi.org/10.1016/j.plantsci.2020.110436.

[5] M. Li, G. Zhou, A. Chen, J. Yi, C. Lu, M. He, and Y. Hu, "FWDGAN-based data augmentation for tomato leaf disease identification," *Computers and Electronics in Agriculture*, vol. 194, p. 106779, Mar. 2022, https://doi.org/10.1016/j.compag.2022.106779.

[6] J. Liu and X. Wang, "Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network," *Frontiers in Plant Science*, vol. 11, p. 898, Jun. 2020, https://doi.org/10.3389/fpls.2020.00898.

[7] K. Marzuki, Apriani, and M. Qulub, "Coffee Roaster Maturity Level Classification Based on Convolutional Neural Network," *Mathematical Modelling of Engineering Problems*, vol. 12, no. 1, pp. 46–54, Jan. 2025, https://doi.org/10.18280/mmep.120106.

[8] J. Liu and X. Wang, "Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model," *Plant Methods*, vol. 16, no. 1, p. 83, Dec. 2020, https://doi.org/10.1186/s13007-020-00624-2.

[9] S. Ahmed, M. B. Hasan, T. Ahmed, M. R. K. Sony, and M. H. Kabir, "Less is More: Lighter and Faster Deep Neural Architecture for Tomato Leaf Disease Classification," *IEEE Access*, vol. 10, pp. 68 868–68 884, 2022, https://doi.org/10.1109/ACCESS.2022.3187203.

[10] T.-H. Nguyen, T.-N. Nguyen, and B.-V. Ngo, "A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease," *AgriEngineering*, vol. 4, no. 4, pp. 871–887, Oct. 2022, https://doi.org/10.3390/agriengineering4040056.

[11] L. Tan, J. Lu, and H. Jiang, "Tomato Leaf Diseases Classification Based on Leaf Images: A Comparison between Classical Machine Learning and Deep Learning Methods," *AgriEngineering*, vol. 3, no. 3, pp. 542–558, Jul. 2021, https://doi.org/10.3390/agriengineering3030035.

[12] M. Sholihin, M. F. M. Fudzee, and M. N. Ismail, "AlexNet-Based Feature Extraction for Cassava Classification: A Machine Learning Approach," *Baghdad Science Journal*, vol. 20, no. 6(Suppl.), p. 2624, Dec. 2023, https://doi.org/10.21123/bsj.2023.9120.

[13] Karthik R., Hariharan M., S. Anand, P. Mathikshara, A. Johnson, and M. R., "Attention embedded residual CNN for disease detection in tomato leaves," *Applied Soft Computing*, vol. 86, p. 105933, Jan. 2020, https://doi.org/10.1016/j.asoc.2019.10593.

[14] M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, "ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network," *Procedia Computer Science*, vol. 167, pp. 293–301, 2020, https://doi.org/10.1016/j.procs.2020.03.225.

[15] N. K. Trivedi, V. Gautam, A. Anand, H. M. Aljahdali, S. G. Villar, D. Anand, N. Goyal, and S. Kadry, "Early Detection and Classification of Tomato Leaf Disease Using High-Performance Deep Neural Network," *Sensors*, vol. 21, no. 23, p. 7987, Nov. 2021, https://doi.org/10.3390/s21237987.

[16] T. Anandhakrishnan and S. Jaisakthi, "Deep Convolutional Neural Networks for image based tomato leaf disease detection," *Sustainable Chemistry and Pharmacy*, vol. 30, p. 100793, Dec. 2022, https://doi.org/10.1016/j.scp.2022.100793.

[17] M. S. Islam, S. Sultana, F. A. Farid, M. N. Islam, M. Rashid, B. S. Bari, N. Hashim, and M. N. Husen, "Multimodal Hybrid Deep Learning Approach to Detect Tomato Leaf Disease Using Attention Based Dilated Convolution Feature Extractor with Logistic Regression Classification," *Sensors*, vol. 22, no. 16, p. 6079, Aug. 2022, https://doi.org/10.3390/s22166079.

[18] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with C-GAN synthetic images," *Computers and Electronics in Agriculture*, vol. 187, p. 106279, Aug. 2021, https://doi.org/10.1016/j.compag.2021.106279.

[19] X. Chen, G. Zhou, A. Chen, J. Yi, W. Zhang, and Y. Hu, "Identification of tomato leaf diseases based on combination of ABCK-BWTR and B-ARNet," *Computers and Electronics in Agriculture*, vol. 178, p. 105730, Nov. 2020, https://doi.org/10.1016/j.compag.2020.105730.

[20] G. Sakkarvarthi, G. W. Sathianesan, V. S. Murugan, A. J. Reddy, P. Jayagopal, and M. Elsisi, "Detection and Classification of Tomato Crop Disease Using Convolutional Neural Network," *Electronics*, vol. 11, no. 21, p. 3618, Nov. 2022, https://doi.org/10.3390/electronics11213618.

[21] M. Badiger and J. A. Mathew, "Tomato plant leaf disease segmentation and multiclass disease detection using hybrid optimization enabled deep learning," *Journal of Biotechnology*, vol. 374, pp. 101–113, Sep. 2023, https://doi.org/10.1016/j.jbiotec.2023.07.011.

[22] P. Baser, J. R. Saini, and K. Kotecha, "TomConv: An Improved CNN Model for Diagnosis of Diseases in Tomato Plant Leaves," *Procedia Computer Science*, vol. 218, pp. 1825–1833, 2023, https://doi.org/10.1016/j.procs.2023.01.160.

[23] S. Yang, L. Zhang, J. Lin, T. Cernava, J. Cai, R. Pan, J. Liu, X. Wen, X. Chen, and X. Zhang, "LSGNet: A lightweight convolutional neural network model for tomato disease identification," *Crop Protection*, vol. 182, p. 106715, Aug. 2024, https://doi.org/10.1016/j.cropro.2024.106715.

[24] M. Melinda, Z. Muthiah, F. Arnia, E. Elizar, and M. Irhmasyah, "Image data acquisition and classification of vannamei shrimp cultivation results based on deep learning," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 23, no. 3, pp. 491–508, 2024, https://doi.org/10.30812/matrik.v23i3.3850.

[This page intentionally left blank.]