Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 3 (2025) 500 - 510 e-ISSN: 2580-0760

Enhancing Tomato Leaf Disease Detection via Optimized VGG16 and Transfer Learning Techniques

Sandy Putra Siregar[1]*, Imam Akbari[2], Poningsih[3], Anjar Wanto[4], Solikhun[5]
[1,2]Informatics Master's Student, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia
[3,4]Department Of Informatics Magister's, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia
[5]Department Informatics Engineering, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia

[1]sandy@amiktunasbangsa.ac.id, [2]imam@amiktunasbangsa.ac.id, [3]poningsih@amiktunasbangsa.ac.id, [4]anjarwanto@amiktunasbangsa.ac.id, [5]solikhun@amiktunasbangsa.ac.id

Abstract

Identification of tomato leaf disease remains difficult because standard approaches are frequently incorrect in identifying distinct signs. Convolutional Neural Networks (CNNs) perform well in image classification and pattern identification, although they are prone to overfitting. Thus, max pooling was employed to reduce dimensionality while retaining crucial information. This paper offers an improved CNN through hyperparameter tuning and compares it to Transfer Learning models such as InceptionV3, NASNetMobile, and VGG16, which were chosen for their efficiency and accuracy. The dataset comprises 7,178 photos classified as Healthy, Leaf Late Blight, Septoria Leaf Spot, and Yellow Leaf Curl Virus, collected from Kaggle.. The dataset is separated into three sections: training, validation, and testing, with a ratio of 70:15:15. The results of this study revealed that the proposed method achieved the highest accuracy of 98.24%. In the application of transfer learning, the inceptionV3 model achieved an accuracy of 96.94%, whereas NASNetMobile obtained 97.50%, and VGG16 showed an accuracy of 96.76%. The evaluation is based on accuracy, precision, recall, F1-score and Inference time to determine the optimum model for accuracy and computing efficiency. This project uses the proposed method and Transfer Learning Techniques to categorize illness images on tomato leaves. These findings will drive further research to improve tehe performance of the proposed method for foliar disease classification and comparable applications.

Keywords: leaf disease; images; classification; proposed method; transfer learning

How to Cite: S. P. Siregar, I. Akbari, P. Poningsih, A. Wanto, and S. Solikhun, "Enhancing Tomato Leaf Disease Detection via Optimized VGG16 and Transfer Learning Techniques", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 9, no. 3, pp. 500 - 510, Jun. 2025.
*Permalink/*DOI: https://doi.org/10.29207/resti.v9i3.6410

Received: February 20, 2025
Accepted: June 8, 2025
Available Online: June 18, 2025

1. Introduction

Diseases such as bacterial spots, early blight, and leaf mold can significantly affect tomato yield. Early detection and treatment of illnesses can help reduce tomato crop losses. Crop disease identification is essential for crop productivity and agricultural production. Deep learning approaches have been the primary research direction for solving the detection of farming diseases[1], [2].

Farmers will benefit more from detecting leaf disease early on and implementing appropriate treatment activities. Thus, an automated system for leaf disease detection that recognizes and classifies leaf diseases at an early stage is needed[3]. As a result, other research by [4] Identifying plant disease is critical for prompt intervention and preventing substantial crop losses. Due to its capacity to identify diseases automatically and accurately, the employment of computer vision technology in phytopathology has risen tremendously.

Tomatoes are one of the most important crops in the world, and they serve as an economic cornerstone in many nations. Identifying plant leaf disease using computer-assisted technology is common nowadays. This study uses CNN and transfer learning to develop a quick and precise method for diagnosing and identifying illnesses on tomato leaves.[5]–[8].

Transfer learning is used in plant village segmentation to retrain the next layer. Furthermore, parameters like epoch and learning rate are adjusted to optimize classification performance[9]. HT+CNN outperforms commonly utilized LCC spectral indices (Sis) regarding LCC estimate reliability. CNN Deep and shallow feature fusion increase the RS LCC soybean estimation ability (Dualex measurement) [10]. Used CNN and Transfer Learning to investigate the identification of COVID-19 in X-ray pictures of human chests. While CNN Scenario 1's accuracy is 95%, Scenario 2's VGG16 93%, Scenario 3's VGG19 is 90%, and Scenario 4's ResNet is 80% [11]. CNN and Transfer Learning algorithms work well for visual object pattern recognition training. This demonstrates that CNN Custom with transfer learning improves plant disease detection accuracy.

The CNN model is optimized with a transfer learning model, which trains the model and transfers knowledge to it. By learning key information from the optimized CNN model and using it in each subsequent cycle, the TL approach improves the training process and increases detection accuracy. [12], [13].The (CNN) and its many transfer learning designs have considerably increased classification performance [14], [15].

This paper describes GA-CNN, a new convolutional neural network model that uses geometric algebra (GA) to preserve inter-channel correlations in hyperspectral picture categorization. The model maintains structural linkages by expressing spectral bands as GA multivectors, exceeding typical CNNs in accuracy and efficiency[16].

Using trained models, The ability of deep image representations to distinguish between features can be significantly improved by integrating the Average Biased ReLU (AB-ReLU) at the network's final layer. AB-ReLU outperformed advanced activation functions like Sigmoid, ReLU, Leaky ReLU, and Flexible ReLU across all seven facial image datasets..[17]. The basic ReLU formula can be seen in Equation 1.

$$f(x) = \max(0, x) \quad (1)$$

This article describes a modified Convolutional Neural Network (CNN) architecture that improves cardiac abnormalities classification by substituting the traditional ReLU activation function with Leaky-ReLU. This tweak enhances feature extraction from 1D ECG data, resulting in an accuracy range of 97%-99% across diverse arrhythmia types[18].

The pooling layer in CNNs is critical for lowering spatial dimensions and increasing computing efficiency. This paper introduces the T-Max-Avg pooling layer, which adaptively picks the K most interacting pixels using a threshold value T, outperforming existing pooling approaches on the CIFAR-10, CIFAR-100, and MNIST datasets[19].

SWM, a CNN accelerator that improves sparse matrix multiplication efficiency using a balanced compressed sparse row structure and dynamic scheduling. Using Winograd adaptability and quantization support, SWM achieves up to 7.6 TOP/s for sparse-Winograd convolution and 3 TOP/s for sparse matrix multiplication on the Xilinx VCU1525 platform[20]. It focuses on optimizing matrix multiplication in convolution layers by combining sparse convolution and weight pruning. The suggested TVM technique optimizes sparsity and speeds up ImageNet-based models by an average of 11.42× compared to the original flow [21].

The high degree of performance demonstrates the use of CNN models for identifying tomato disease and this method. An integrated system for identifying various plant diseases may be possible by extension. A convolutional neural network-based deep learning model is employed to categorize cement paste mixtures based on varying water-cement ratios and additional silica fume content..[22], [23].

To develop the best model, CNN models are optimized by analyzing various hyperparameters, such as optimizers and learning rates, and comparing model performance on concrete defect classification[24]. Custom CNN models with hyperparameters are highly beneficial for conducting research, introducing hyperparameter modifications, and increasing knowledge for readers and future studies. Feature extraction is performed using pre-trained weights from MobileNetV2 and NASNetMobile architectures.. The extracted features are combined and their dimensionality is reduced through kernel principal component analysis (KPCA), before being input into a conventional machine learning algorithm[25].

This paper describes a novel approach to content-based image retrieval (CBIR) that combines deep learning models (NASNerMobile, DenseNet121, and VGG16) to increase search accuracy and relevance by using adaptability via hard and soft voting strategies [26]. This study compares NASNet-Mobile-based transfer learning models to custom convolutional neural network (CNN) topologies, demonstrating that transfer learning improves model accuracy by 5% on average and reduces loss by 15%. Additionally, experiments show that fine-tuning can significantly improve the performance of custom CNN models[27].

Convolutional Neural Network is a deep learning technique that can train enormous datasets, create a two-dimensional visual representation, and combine it with various filters to generate the desired result. Deep Learning includes methods such as CNN. As a result, medical problems, particularly those involving image recognition in Alzheimer's disease, can be solved by utilizing a deep-learning technique[28]. The study recognized each character in an antique document using an enhanced OCR system based on convolutional neural networks (CNNs). A group of convolutional layers in the suggested system are used to extract deep hierarchical feature vectors from the input character image. Two fully connected neural network (FCNN) layers have classified these feature vectors into the appropriate class[29]. This study uses deep learning approaches to assess and optimize multiple pre-trained deep learning models on ImageNet datasets to detect tomato leaf disease [30]. One of the pre-trained models is quite helpful and lightweight when implemented using ImageNet.

This study employs an optimized CNN model to enhance classification accuracy. The model's design includes convolutional layers to extract features and fully connected layers for classification, while activation functions are applied to introduce nonlinearity at various stages of the network.[31], [32]. The CNN model, VGG, is used in this study, "Transfer Learning For Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG". The primary purpose of this research is to develop a model that will increase the accuracy of diseased leaf categorization results. In experiments, the proposed study achieved 98.40% accuracy for grape leaf classification and 95.71% for tomato leaf classification [33]. In this study, the usage of the Inception-V3 algorithm on CNN demonstrates that the accuracy obtained is suboptimal and can be improved. The hyperparameters can yet be improved

In the Previous study, Tomato Disease Through Leaf Image using Convolutional Neural Network Method, detection models were implemented utilizing two types of Convolutional Neural Network architectures: regular LeNet-5 and modified LeNet-5 (custom). The normal LeNet-5 reliably identifies tomato leaves with 90% accuracy. However, the modified LeNet-5 architecture in the network layer and parameters achieved 95% accuracy [34]. In this research, increasing the data can still improve the accuracy of the Proposed Method.

The two earlier studies had some flaws, most notably the failure to optimize the model's hyperparameters. As a result, the researchers conducted this study using experiments designed to improve model performance.

A strategy was developed to address issues identified in previous research, such as those involving Proposed Methods, based on various data balancing methods and hyperparameter adjustments. The data splitting technique partitioned the dataset, enabling diverse training variations. Meanwhile, hyperparameter tuning was utilized to enhance model performance and prevent overfitting during training. The data splitting technique also helped balance majority and minority class data, thereby improving accuracy by avoiding overfitting. Previous studies have demonstrated that Proposed Methods are more effective in resolving these challenges.

Transfer learning is a technique for solving new problems using pre-trained CNN models on massive datasets like ImageNet. This technique reuses the weights from the pre-trained model on a new dataset, reducing the requirement to train from scratch. Transfer learning reduces training time, improves model performance, and is appropriate for application on small datasets. These benefits make it a valuable research tool that demands quick forecasts and excellent accuracy.

This study uses three models to train data: Proposed Method, Inception-V3 Transfer Learning, and NasNetMobile Transfer Learning. The model uses pre-trained weight. In conclusion, this study describes the implementation of a Proposed Method with Transfer Learning to evaluate and compare performance using leaf disease detection data from tomato plants. With knowledge of the development of artificial intelligence and pattern recognition, these findings will assist in making the best choice for the perfect diagnosis of tomato leaf disease..

2. Methods

2.1 Research Procedures

This study proposes a CNN optimized through hyperparameter tuning and compares it with Transfer Learning models: InceptionV3, which uses factorized convolutions for efficiency; NASNetMobile, designed for low-computation devices; and VGG16, a simple but effective architecture for feature extraction. The following research procedure is shown in Figure 1.

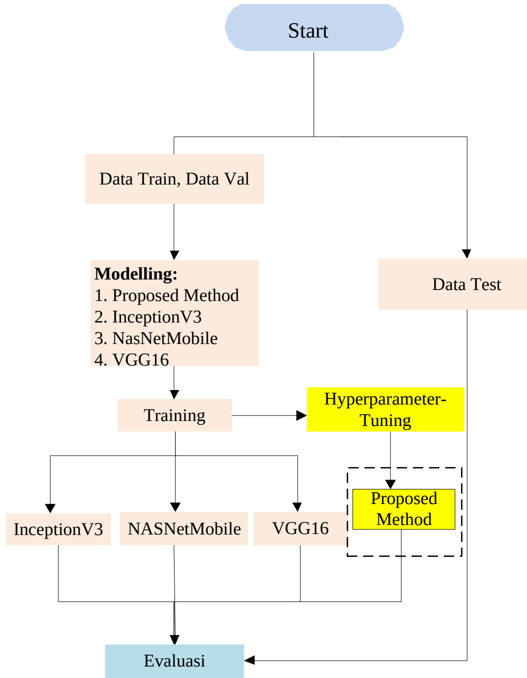
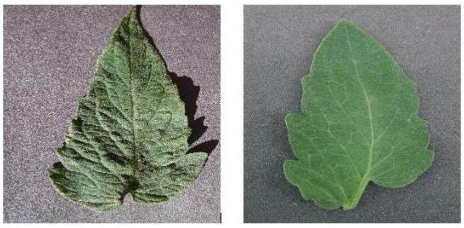


Figure 1. Research Procedures

Figure 1 lists the procedures for this study. This research begins with acquiring a dataset containing photographs of tomato leaves affected by various illnesses, which is then preprocessed. The preprocessing stage begins with balancing the dataset as the initial step in a multi-phase process. The following step after preprocessing is to process the data clustering findings. At this stage, the dataset is divided into three parts: 70% for training data, 15% for validation data, and 15% for testing data. With this division, the training data covers 70% of the 7,178 images in the dataset, the validation data covers 15% of the total, and the testing data covers 15% of the total. This division ensures that the model can be trained, validated, and tested optimally. In the train data, a modeling stage is used to train the model using predefined or custom hyperparameters. To evaluate potential overfitting or underfitting, model performance is first examined on validation data. After training, the model is tested on unseen test data to measure its effectiveness. The final phase involves a thorough evaluation to ensure the model is stable, avoids overfitting, and generalizes well to new data. generalization means that the model works with the training data and is more varied in prediction to be more accurate with similar patterns.

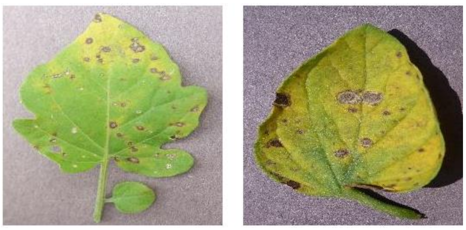
2.2 Dataset

The dataset comprises 7178 images of four types of leaves: Healthy, leaf late blight, Septoria leaf spot, and yellow leaf curl virus obtained from Kaggle. depicts 7,178 images divided into four categories: healthy leaves (1,591 images), late blight-infected leaves (1,909 images), septoria leaf spot-infected leaves (1,771 images), and yellow leaf curl virus-infected leaves (1,907 images).



(a) Leaf Healthy

(b) Leaf Late Blight



(c) Septoria Leaf Spot

(d) Yellow Leaf Curl Virus

Figure 2. Sample Dataset (a) Leaf Healthy, (b) Leaf Late Blight, (c) Septoria Leaf Spot, and (d) Yellow Leaf Curl Virus

The dataset is divided into three categories: 70% training data, 15% validation data, and 15% test data from the total data in each class. The quantity of datasets before the division process remains unchanged, with no additional images added. The images are 256×256 pixels and in JPG format. Figure 2 illustrate example image datasets of each class.

2.3 Proposed Method Architecture

The proposed method uses a modified VGG16 architecture or hyperparameter tuning, with an image size of 224x24 pixels set as the input layer. This research chose a batch size of 64, commonly used for large-scale training. Figure 3 shows the architecture of the proposed method and vgg16 baseline.

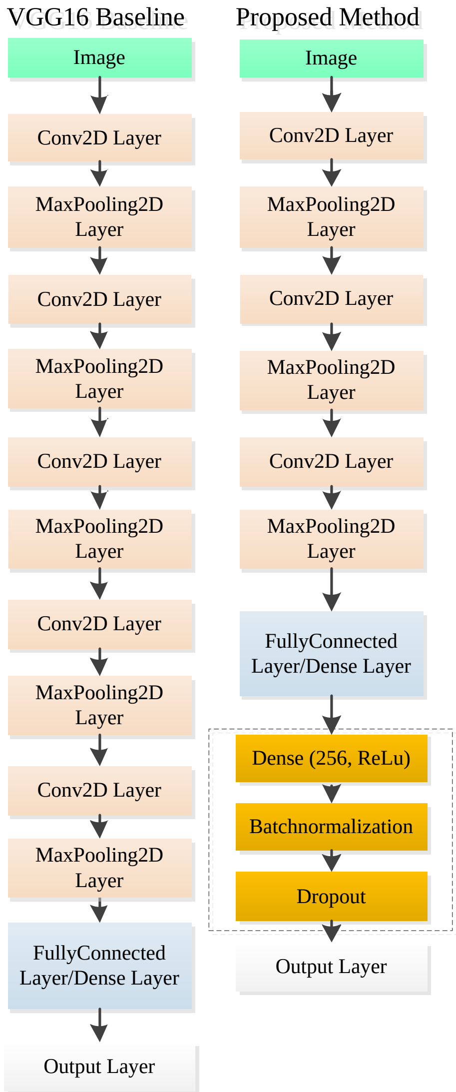


Figure 3. Proposed Method Architecture Model

The sequence in Figure 3 depicts the architecture of the first proposed method. The model is made up of three basic, each with a convolution layer (Conv2D) followed by a pooling layer (MaxPooling2D). The convolution layer extracts essential elements from the image, such as patterns and textures, while the pooling layer minimizes the dimensionality of the data and maintains the most relevant information, making the model more efficient. The pooling layer employs the maximum pooling method with 3 x 3 filters. In addition, convolutional layers are constructed with 128, 64, and 32 filter counts. Each of these layers has a 3 x 3 kernel and is activated by ReLU (Rectified Linear Unit). This study employs a CNN-based model optimized using the Adam algorithm, configured with a learning rate of 0.0001.

2.4 Hyperparameter Tuning

To optimize the model, this work uses hyperparameter-tuning to alter the major parameters in the dense layer. To avoid overfitting and enhance efficiency, performance evaluation determined that the appropriate number of neurons in the first dense layer was 256. ReLU accelerates convergence and overcomes vanishing gradients, whereas Batch Normalization stabilizes activation and speeds up training. Dropout Rate 0.5 was used to reduce overfitting while maintaining accuracy. Adam's learning rate of 0.0001 was the most effective in balancing convergence and accuracy. The dashed gray box line in the figure above represents this study's innovation: systematic hyperparameter-tuning to improve classification efficiency. The last dense layer generates each class's probability distribution using Softmax. The following hyperparameter values are shown in Table 1.

Table 1. Hyperparameter Values

Parameter	Comparative Value
Dense Layer	256 (ReLu)
Batchnormalization	-
Dropout	0,5
Optimizer	Adam

2.5 Evaluation

A technique for evaluating a classification model’s performance is the confusion matrix. There are four parts to the confusion matrix: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Using these numbers, we can figure out how well classification models work by looking at metrics like accuracy, precision, recall, and F1-score. Equation 2 shows that accuracy is the rate at which predictions are correct overall. In Equation 3, precision shows how well the model avoids giving false positives. Equation 4 checks recall, which shows how well the model can find all the relevant positive cases. Lastly, Equation 5 says that the F1-score is the harmonic mean of precision and recall.[35]. Observed in Equations 2 through 5.

$$Accuracy = \frac{TP}{TP+TN+FP+FN} \times 100\% \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (4)$$

$$F1 - score = \frac{FP}{TP} \times 100\% \quad (5)$$

3. Results and Discussions

The findings of this study include the steps performed according to the test setup. Tests were run on the four proposed models, and their performance was compared. Each model was tested to determine its performance, and the results were compared using the performance factors of accuracy, precision, and recall. In addition, The next stage is model building and training, which requires 20 iterations (epochs) to achieve optimal results.

3.1 Results

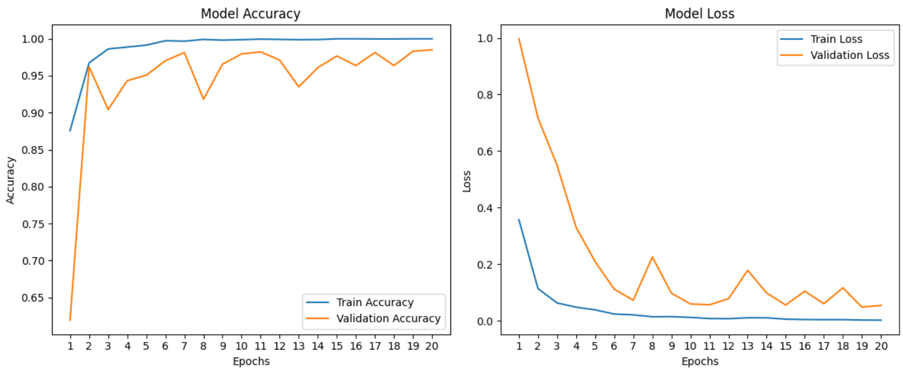
In Figure 4, the training process is controlled by monitoring the loss and accuracy values. Figure 4 shows the proposed method results from changing the CNN architecture, namely VGG16, with hyperparameter tuning such as dropout, regularization, batchnormalization, and others. Training with the proposed method results in a plot graph depicting the accuracy and loss graphs illustrated in Figures 4 (a). The results of the data training are shown in the second plot graph, which has an accuracy value of 0.9824 (98.24%) and a loss value of 0.0705.

The results of the model training are shown in the second graph, with an accuracy of 0.9824 (98.24%), indicating the model's performance in predicting the data with a high degree of success. The loss value of 0.0705 indicates that the error between the model's prediction results and the actual data is quite small, reflecting the model's ability to effectively understand the data pattern. The combination of high accuracy and low loss indicates that the model is well trained, although it needs to be further analyzed to ensure there is no overfitting.

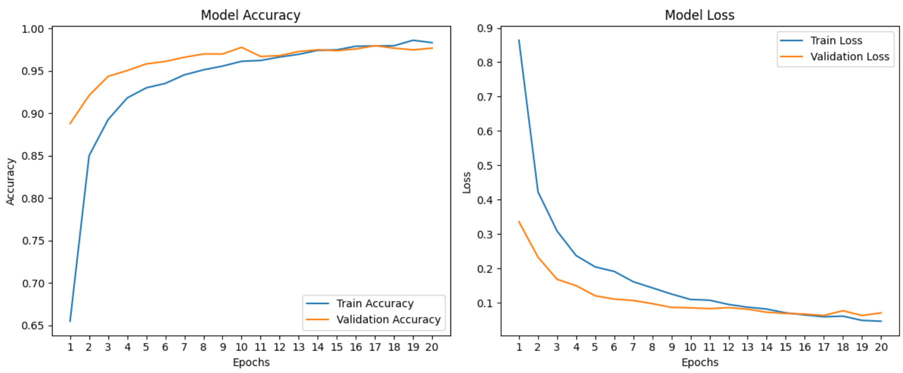
InceptionV3 is a transfer learning model with a CNN architecture that includes hyperparameters like dropout and regularization. The training produced a graphical plot depicting the accuracy and loss graphs seen in Figures 4 (b). The second graphical figure displays the data training outcomes, with an accuracy value of 0.9694 (96.94%) and a loss value of 0.0980. High accuracy paired with low loss suggests the model has been effectively trained; however, further examination is necessary to confirm that overfitting has not taken place.

NASNetMobile is a transfer learning model with CNN architecture that includes hyperparameters such as batchnormalization, dropout and regularization. The training resulted in graphical plots depicting the accuracy and loss graphs seen in Figures 4 (c). The second graphical image displays the results of the training data, with an accuracy value of 0.9710 (97.10%) and a loss value of 0.0817.

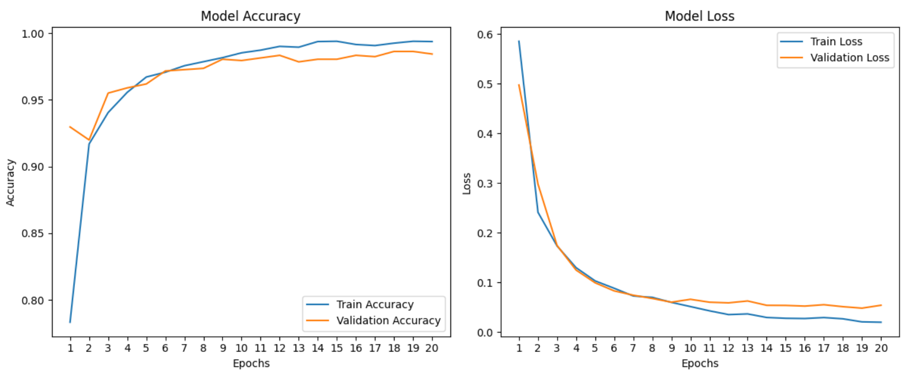
This study also used a learning rate of 0.0001. The results of the model training are shown in the second graph, with an accuracy of 0.9710 (97.10%), indicating the model's performance in predicting the data with a high degree of success.



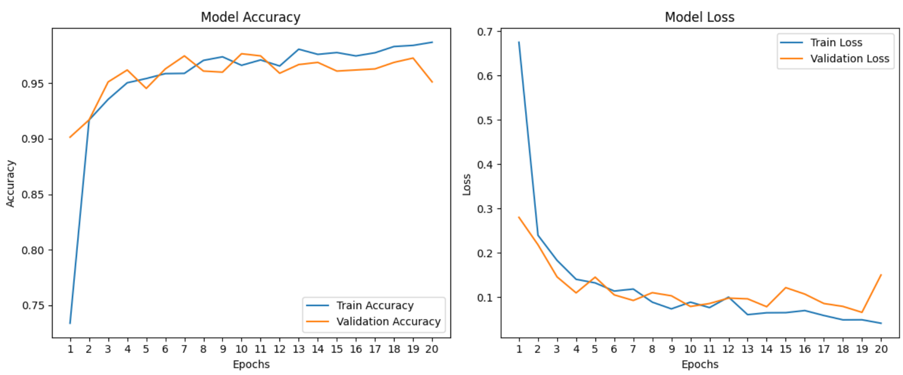
(a) Proposed method



(b) InceptionV3



(c) NASNetMobile



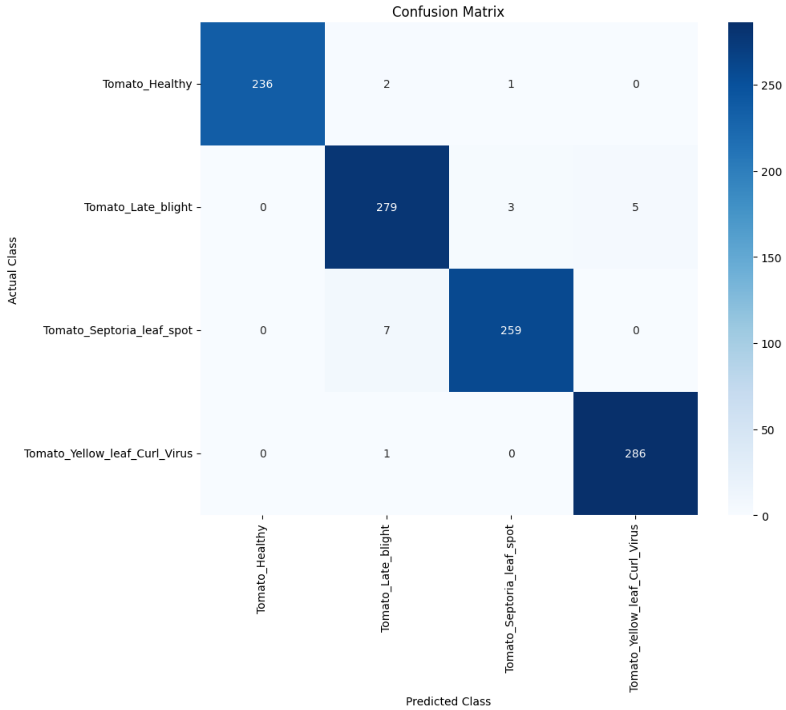
(d) VGG16

Figure 4. Results accuracy and loss training for (a) Proposed Method, (b) InceptionV3, (c) NASNetMobile, (d) VGG16

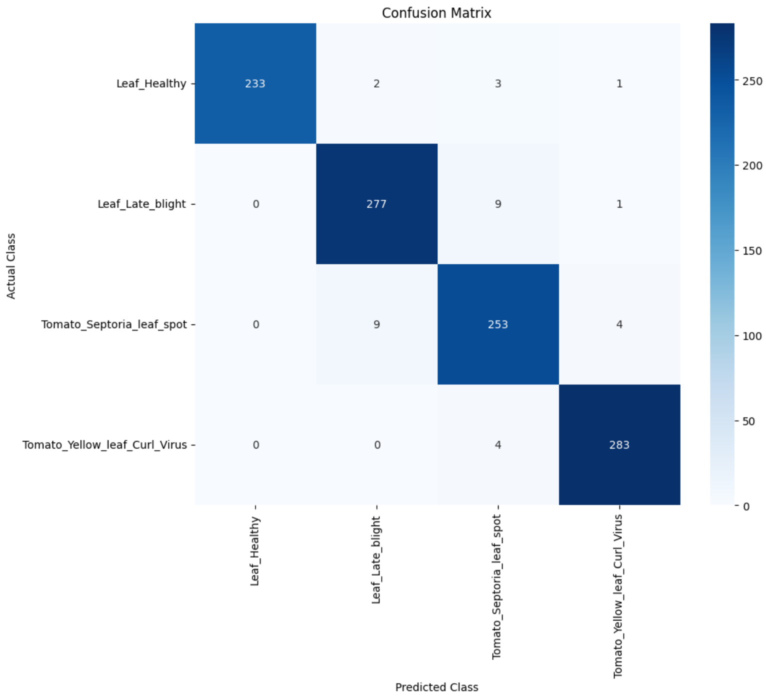
A loss of 0.0817 indicates a minimal discrepancy between the model's predictions and the actual data, demonstrating effective pattern recognition. While high accuracy combined with low loss suggests successful training, further evaluation is needed to ensure the model hasn't overfit the data.

VGG16 is a transfer learning model with a CNN architecture that includes hyperparameters such as dropout to keep the data from overfitting. The training resulted in graphical plots depicting the accuracy and loss graphs in Figure 4 (d). The second graphical image displays the training data results, with an accuracy value of 0.9676 (97.76%) and a loss value of 0.1245.

Figures 5 and 6 illustrate further model evaluation using a confusion matrix.

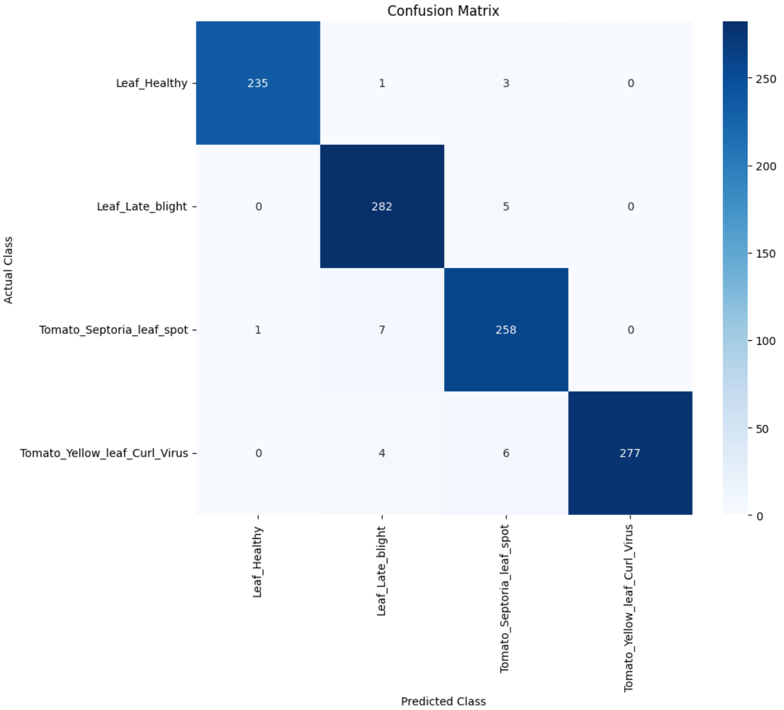


(a). Proposed Method

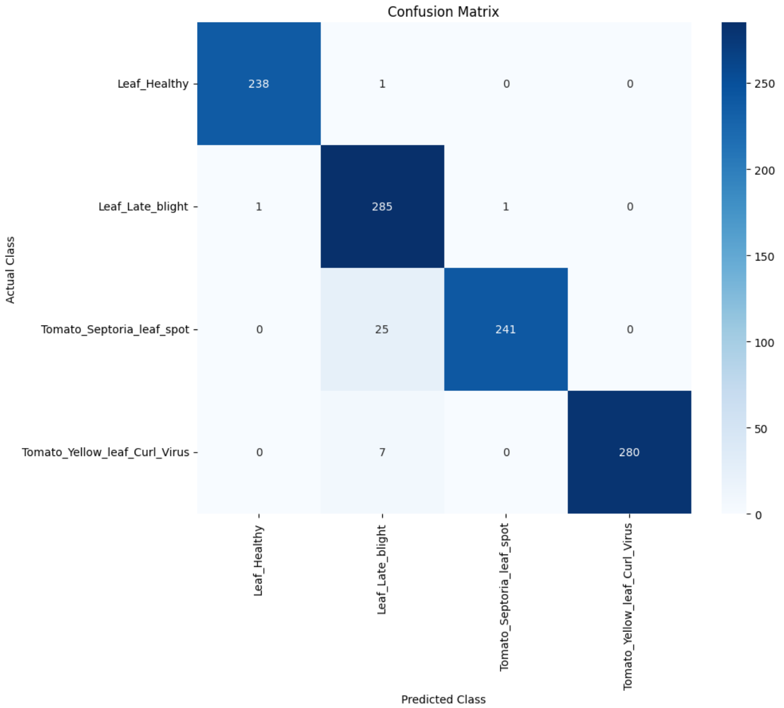


(b). InceptionV3

Figure 5. Confusion matrix proposed method and inceptionV3



(a). NASNetMobile



(b). VGG16

Figure 6. Confusion matrix NASNetMobile and VGG16

The Figure 5 proposed method (a) confusion matrix demonstrates that in the healthy leaf class, 233 image data are correctly predicted and 7 image data are wrongly predicted. In the class of leaves infected by late blight disease, 279 image data are predicted properly, whereas 5 are projected inaccurately. In the septoria leaf spot class, 228 image data are predicted correctly, whereas 9 are projected inaccurately. For the yellow leaf curl virus class, 286 image data are successfully predicted, whereas 6 are wrongly predicted.

InceptionV3 (b) The confusion matrix demonstrates that 233 image data are correctly predicted in the healthy leaf class, and 7 are wrongly predicted. In the leaves infected by late blight disease class, 279 image data are correctly predicted, whereas 5 are projected inaccurately. In the septoria leaf spot class, 228 image data are predicted correctly, whereas 9 are projected inaccurately. For the yellow leaf curl virus class, 286 image data are successfully predicted, whereas 6 are wrongly predicted.

Figure 6 NASNetMobile (left) demonstrates that 4 photos in the healthy leaf class were mispredicted, whereas 235 images were correctly assigned. There were 282 photos found in the late blight-infected leaf class, and 5 images were mispredicted. There were 7 wrongly predicted images and 229 properly predicted images in the Septoria leaf spot class. There were 6 wrongly predicted photos and 277 properly predicted images in the yellow leaf curl virus class.

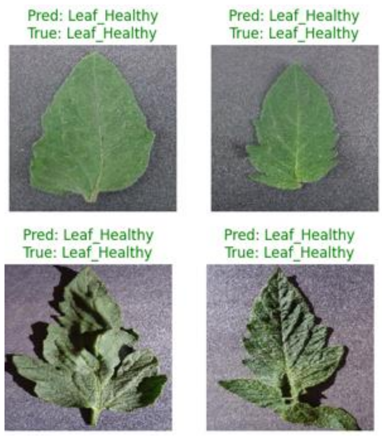
VGG16 (b) after analysis shows, leaf healthy class has 238 correct predictions and only 1 error, while leaf late blight has 285 correct predictions with 2 errors. Meanwhile, tomato Septoria leaf spot had 25 misclassifications, with a total of 241 correct predictions. The tomato yellow leaf curl virus class contained 280 correct predictions and 7 errors, mainly classified as leaf late blight.

3.2 Image Classification Prediction Results

Next, prediction results from the proposed models, InceptionV3, NASNetMobile, and VGG16, were tested on test data presenting 10 images, but 4 prediction images were taken, resulting in an average of correct predictions. Figure 7 illustrates the prediction results.

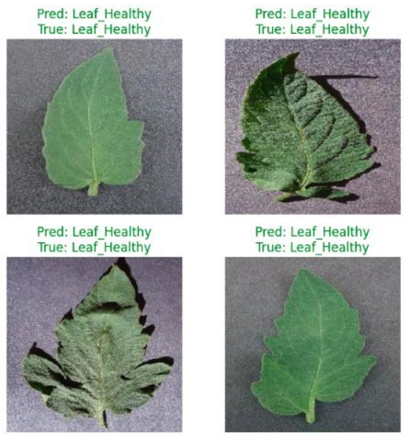
Figure 7, The proposed model demonstrates that all four picture samples from each class were accurately identified when expected. InceptionV3 successfully classified all four data, demonstrating that this model performs well in prediction. NASNetMobile accurately classified all four picture samples. Meanwhile, VGG16 had the lowest accuracy, predicting only three samples correctly and one inaccurately. The sample findings demonstrate that the models can produce accurate forecasts with minimum mistakes, particularly the proposed model. Furthermore, the sample results show that the suggested method has a faster and more efficient prediction time than existing models.

2/2 [==============================] - 1s 632ms/step



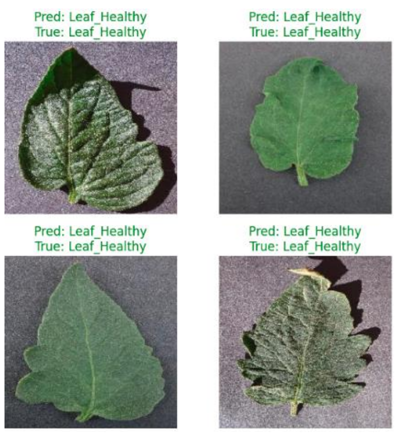
(a)Proposed Method

2/2 [==============================] - 6s 2s/step



(b) InceptionV3

2/2 [==============================] - 8s 2s/step



2/2 [==============================] - 19s 9s/step

(a) NASNetMobile (b) VGG16

Figure 7. Test Prediction Results with Inference Time (a) Proposed Method, (b) InceptionV3, (c) NASNetMobile, and (d) VGG16

3.3 Evaluation of Results

The outcomes demonstrate the suggested method's superiority in managing intricate datasets and effectively capturing intricate features. It uses Inception's module-based architecture, which enables efficient multi-scale feature extraction by combining convolution kernels of different sizes. In addition, NASNetMobile comes as a model designed with Neural Architecture Search (NAS) techniques to optimize the balance between accuracy and efficiency, making it a good choice for resource-constrained devices. On the other hand, VGG16, despite having a more profound architecture and being able to capture complex features well, tends to require more computing power, making it less efficient than more modern models such as NASNetMobile or the proposed method. Table 2 Classification results from training in this study were analyzed using evaluation metrics such as precision, recall, F1-score, and test accuracy.

Table 2. Results of precision, recall, F1-score and Accuracy

	Precision	Recall	F1-Score	Accuracy
Proposed Method	98,25%	98,25%	98,25%	98,25%
InceptionV3	97%	97%	97%	97%
NASNetMobile	97,75%	97,5%	97,5%	97,5%
VGG16	97,5%	97%	97%	97%

Table 2 shows that, based on the classification results, the Proposed Method performed best with precision, recall, F1-score, and accuracy of 98.25%, outperforming the other models in correctly recognizing the samples. InceptionV3 recorded 97% on all metrics, showing a good balance but still lower than the Proposed Method. NASNetMobile had a precision of 97.75% and recall of 97.5%, with an F1-score and accuracy of 97.5%, performing little better than InceptionV3. In contrast, VGG16 has 97% accuracy, 97% recall, 97% F1-score, and 97.5% precision., equivalent to InceptionV3 but still inferior to NASNetMobile and Proposed Method. Overall, these results show that the Proposed Method is superior in image classification with higher accuracy and balanced metrics, proving the effectiveness of the optimization applied in this study. Figure 8 shows the evaluation diagram for comparison with the values in Table 2.

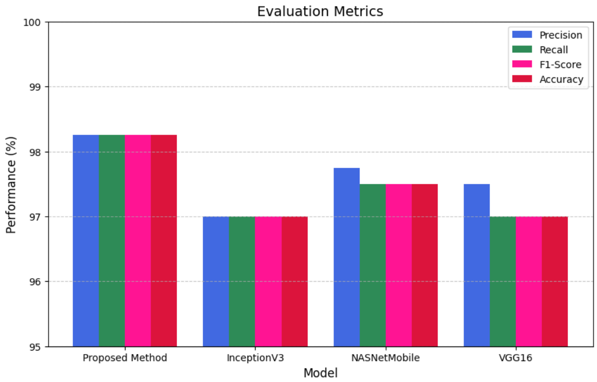


Figure 8. Comparison Evaluation Metric

4. Conclusions

This study emphasizes the necessity of selecting hyperparameter adjustments in CNN architecture, notably for classifying tomato plant leaf diseases. This study compares hyperparameter change and transfer learning with InceptionV3, NASNetMobile, and VGG16 using the improved Proposed Method. The findings reveal a precision of 98.25%, recall of 98.25%, F1-score of 98.25%, and accuracy of 98.25%, indicating that the proposed strategy effectively deals with this categorization difficulty.

Additionally, the results demonstrate that choosing the right hyperparameters is essential for maximizing F1-score, accuracy, precision, and recall. However, this study has limitations, such as using a dataset of 7178 which may still be limited in the diversity of leaf disease images. Therefore, future research should explore more significant and diverse datasets and test them with ensemble models and other configurations of hyperparameter shapes.

Acknowledgements

The writers say they have no competing interests.

References

[1] 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, 2021, doi: 10.3390/agriengineering3030035.

[2] S. Zhao, Y. Peng, J. Liu, and S. Wu, “Tomato leaf disease diagnosis based on improved convolution neural network by attention module,” *Agric.*, vol. 11, no. 7, 2021, doi: 10.3390/agriculture11070651.

[3] T. Anandhakrishnan and S. M. Jaisakthi, “Deep Convolutional Neural Networks for image based tomato leaf disease detection,” *Sustain. Chem. Pharm.*, vol. 30, 2022, doi: 10.1016/j.scp.2022.100793.

[4] 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,” *Agric.*, vol. 12, no. 2, 2022, doi: 10.3390/agriculture12020228.

[5] R. Thangaraj, S. Anandamurugan, and V. K. Kaliappan, “Automated tomato leaf disease classification using transfer learning-based deep convolution neural network,” *J. Plant Dis. Prot.*, vol. 128, no. 1, pp. 73–86, 2021, doi: 10.1007/s41348-020-00403-0.

[6] O. Attallah, “Tomato Leaf Disease Classification via Compact Convolutional Neural Networks with Transfer Learning and Feature Selection,” *Horticulturae*, vol. 9, no. 2, 2023, doi: 10.3390/horticulturae9020149.

[7] P. Kaur, S. Harnal, V. Gautam, M. P. Singh, and S. P. Singh, “An approach for characterization of infected area in tomato leaf disease based on deep learning and object detection technique,” *Eng. Appl. Artif. Intell.*, vol. 115, 2022, doi: 10.1016/j.engappai.2022.105210.

[8] S. G. Paul *et al.*, “A real-time application-based convolutional neural network approach for tomato leaf disease classification,” *Array*, vol. 19, 2023, doi: 10.1016/j.array.2023.100313.

[9] 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, 2022, doi: 10.3390/agriengineering4040056.

[10] J. Yue *et al.*, “Hyperspectral-to-image transform and CNN

transfer learning enhancing soybean LCC estimation," *Comput. Electron. Agric.*, vol. 211, p. 108011, 2023, doi: 10.1016/j.compag.2023.108011.

[11] R. Mostafiz, M. S. Uddin, N. A. Alam, M. Mahfuz Reza, and M. M. Rahman, "Covid-19 detection in chest X-ray through random forest classifier using a hybridization of deep CNN and DWT optimized features," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3226–3235, 2022, doi: 10.1016/j.jksuci.2020.12.010.

[12] C. Liu *et al.*, "A transfer learning CNN-LSTM network-based production progress prediction approach in IIoT-enabled manufacturing," *Int. J. Prod. Res.*, vol. 61, no. 12, pp. 4045–4068, 2023, doi: 10.1080/00207543.2022.2056860.

[13] U. K. Lilhore *et al.*, "HIDM: Hybrid Intrusion Detection Model for Industry 4.0 Networks Using an Optimized CNN-LSTM with Transfer Learning," *Sensors*, vol. 23, no. 18, 2023, doi: 10.3390/s23187856.

[14] P. Agarwal, A. Yadav, and P. Mathur, "Breast Cancer Prediction on BreakHis Dataset Using Deep CNN and Transfer Learning Model," *Lect. Notes Networks Syst.*, vol. 238, pp. 77–88, 2022, doi: 10.1007/978-981-16-2641-8_8.

[15] L. Dong, H. Shu, Z. Tang, and X. Yan, "Microseismic event waveform classification using CNN-based transfer learning models," *Int. J. Min. Sci. Technol.*, vol. 33, no. 10, pp. 1203–1216, 2023, doi: 10.1016/j.ijmst.2023.09.003.

[16] Y. Li *et al.*, "GA-CNN: Convolutional Neural Network Based on Geometric Algebra for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, doi: 10.1109/TGRS.2022.3212682.

[17] S. R. Dubey and S. Chakraborty, "Average biased ReLU based CNN descriptor for improved face retrieval," *Multimed. Tools Appl.*, vol. 80, no. 15, pp. 23181–23206, 2021, doi: 10.1007/s11042-020-10269-x.

[18] K. Lakhdari and N. Saeed, "A new vision of a simple 1D Convolutional Neural Networks (1D-CNN) with Leaky-ReLU function for ECG abnormalities classification," *Intell. Med.*, vol. 6, 2022, doi: 10.1016/j.ibmed.2022.100080.

[19] L. Zhao and Z. Zhang, "A improved pooling method for convolutional neural networks," *Sci. Rep.*, vol. 14, no. 1, pp. 1–22, 2024, doi: 10.1038/s41598-024-51258-6.

[20] D. Wu, X. Fan, W. Cao, and L. Wang, "SWM: A High-Performance Sparse-Winograd Matrix Multiplication CNN Accelerator," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 29, no. 5, pp. 936–949, 2021, doi: 10.1109/TVLSI.2021.3060041.

[21] H. H. Liao, C. L. Lee, J. K. Lee, W. C. Lai, M. Y. Hung, and C. W. Huang, "Support Convolution of CNN with Compression Sparse Matrix Multiplication Flow in TVM," *ACM Int. Conf. Proceeding Ser.*, 2021, doi: 10.1145/3458744.3473352.

[22] A. Saeed, A. A. Abdel-Aziz, A. Mossad, M. A. Abdelhamid, A. Y. Alkhaled, and M. Mayhoub, "Smart Detection of Tomato Leaf Diseases Using Transfer Learning-Based Convolutional Neural Networks," *Agric.*, vol. 13, no. 1, 2023, doi: 10.3390/agriculture13010139.

[23] H. Qian, Y. Li, J. Yang, L. Xie, and K. H. Tan, "Image-based microstructure classification of mortar and paste using convolutional neural networks and transfer learning," *Cem. Concr. Compos.*, vol. 129, p. 104496, 2022, doi: 10.1016/j.cemconcomp.2022.104496.

[24] P. Arafin, A. Issa, and A. H. M. M. Billah, "Performance Comparison of Multiple Convolutional Neural Networks for Concrete Defects Classification," *Sensors*, vol. 22, no. 22, p. 8714, 2022, doi: 10.3390/s22228714.

[25] M. S. A. M. Al-gaashani, F. Shang, M. S. A. Muthanna, M. Khayyat, and A. A. Abd El-Latif, "Tomato leaf disease classification by exploiting transfer learning and feature concatenation," *IET Image Process.*, vol. 16, no. 3, pp. 913–925, 2022, doi: 10.1049/ipr2.12397.

[26] S. S. Sadiq, "Improving CBIR Techniques with Deep Learning Approach: An Ensemble Method Using NASNetMobile, DenseNet121, and VGG12," *J. Robot. Control*, vol. 5, no. 3, pp. 863–874, 2024, doi: 10.18196/jrc.v5i3.21805.

[27] I. Naskinova, "Transfer learning with NASNet-Mobile for Pneumonia X-ray classification," *Asian-European J. Math.*, vol. 16, no. 1, p. 2250240, 2023, doi: 10.1142/S1793557122502400.

[28] A. Shukla, R. Tiwari, and S. Tiwari, "Alz-ConvNets for Classification of Alzheimer Disease Using Transfer Learning Approach," *SN Comput. Sci.*, vol. 4, no. 4, 2023, doi: 10.1007/s42979-023-01853-7.

[29] A. Jindal and R. Ghosh, "An optimized CNN system to recognize handwritten characters in ancient documents in Grantha script," *Int. J. Inf. Technol.*, vol. 15, no. 4, pp. 1975–1983, 2023, doi: 10.1007/s41870-023-01247-1.

[30] H. Tarek, H. Aly, S. Eisa, and M. Abul-Soud, "Optimized Deep Learning Algorithms for Tomato Leaf Disease Detection with Hardware Deployment," *Electron.*, vol. 11, no. 1, p. 140, 2022, doi: 10.3390/electronics11010140.

[31] N. N. Prakash, V. Rajesh, D. L. Namakhwa, S. Dwarkanath Pande, and S. H. Ahammad, "A DenseNet CNN-based liver lesion prediction and classification for future medical diagnosis," *Sci. African*, vol. 20, 2023, doi: 10.1016/j.sciaf.2023.e01629.

[32] W. N. Ismail, H. A. Alsalamah, M. M. Hassan, and E. Mohamed, "AUTO-HAR: An adaptive human activity recognition framework using an automated CNN architecture design," *Heliyon*, vol. 9, no. 2, p. e13636, 2023, doi: 10.1016/j.heliyon.2023.e13636.

[33] A. S. Paymode and V. B. Malode, "Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG," *Artif. Intell. Agric.*, vol. 6, pp. 23–33, 2022, doi: 10.1016/j.aiia.2021.12.002.

[34] R. H. Saputra, R. C. S. Hariyono, and F. Fathulloh, "Deteksi Penyakit Tomat melalui Citra Daun menggunakan Metode Convolutional Neural Network," *Avitec*, vol. 5, no. 1, p. 43, 2023, doi: 10.28989/avitec.v5i1.1404.

[35] M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: Multi-Label Confusion Matrix," *IEEE Access*, vol. 10, pp. 19083–19095, 2022, doi: 10.1109/ACCESS.2022.3151048.