Jurnal Masyarakat Informatika, 16(2), 2025
DOI:10.14710/jmasif.16.2.73168
ISSN: 2777-0648

Modified VGG16 Architecture with Bayesian Optimization for Tomato Leaf Disease Classification

Tsaqif Muhammad Arkan, Aris Sugiharto, and Helmie Arif Wibawa[*]
Department of Informatics, Diponegoro University, Semarang, Indonesia
[*] *Corresponding author: helmie.arif@live.undip.ac.id*

Abstract
This study proposes an optimized VGG16 architecture enhanced through Bayesian Optimization to improve the classification of tomato leaf diseases. The modified model integrates tunable parameters such as dropout rates, convolutional filters, and dense units, while maintaining the foundational structure of VGG16. To further refine performance, Bayesian Optimization is employed to search for the most effective combination of hyperparameters. Experiments conducted using the Tomato Leaf Disease Detection dataset demonstrate that the proposed method outperforms the original VGG16 model, achieving a test accuracy of 97.1% compared to 89.0%. These results underline the importance of architecture customization and systematic hyperparameter tuning for domain-specific deep learning tasks in agriculture.

Keywords : *Tomato leaf disease, VGG16, Bayesian Optimization, Deep learning, Image classification, Hyperparameter tuning*

1 Introduction

Tomato (Solanum lycopersicum) is among the most significant horticultural crops grown worldwide because of its nutritional benefits and high economic value. However, tomato plants are susceptible to various foliar diseases caused by pathogens such as fungi, bacteria, viruses, and pests. These diseases severely affect yield and fruit quality, highlighting the necessity for early and accurate disease detection to ensure sustainable agricultural production.

Traditional methods of disease detection rely on manual inspection by agricultural experts, which are labor-intensive, time-consuming, and often subject to human error and inconsistencies. Recent advancements in artificial intelligence (AI), particularly in deep learning, have made the automation of plant disease detection through image classification increasingly feasible. Convolutional Neural Networks (CNNs) have shown state-of-the-art performance in various visual classification tasks, including identifying crop diseases. Among the different CNN architectures, VGG16 [1] is widely recognized for its uniform 3×3 convolution kernels and its deep architecture. Its effectiveness in general image classification has also resulted in strong performance in agricultural applications, particularly in plant disease diagnosis. However, the original VGG16 architecture contains approximately 138 million parameters, making it vulnerable to overfitting and unsuitable for lightweight or resource-constrained applications, such as those used in the field or on mobile devices [2].

Several studies have attempted to address these challenges by optimizing the VGG16 architecture. This includes architectural modifications such as the integration of Inception modules [3], attention mechanisms [4], and the use of Global Average Pooling [5]. These enhancements not

Received: 12 May 2025 ; Revised: 12 June 2025; Accepted: 16 June 2025; Published: 18 June 2025

only reduce the number of trainable parameters but also improve feature extraction and classification performance. Equally important is the selection of optimal hyperparameters, which significantly influences model training dynamics and final performance. Traditional methods such as grid search and manual tuning are computationally inefficient. Bayesian Optimization offers a promising alternative by modeling the objective function probabilistically and using acquisition functions to efficiently explore the hyperparameter space [6]. Recent studies by Khan et al. [7] and Khan et al. [4] demonstrate that applying Bayesian Optimization to CNN architectures can lead to significant gains in classification accuracy while reducing training time and mitigating overfitting.

The rest of this paper is organized as follows. Section 2 reviews related work in plant disease classification, use of VGG16, and hyperparameter optimization. Section 3 describes the proposed methodology, including dataset preprocessing, architectural changes, and optimization strategy. Section 4 presents the results and analysis. Finally, Section 5 concludes the paper and outlines directions for future research.

2 Literature Review

Recent years have seen a surge in the application of deep learning techniques for agricultural diagnostics, particularly in plant disease identification. One of the foundational works by Mohanty et al. [8] employed AlexNet and GoogLeNet to classify 26 diseases across 14 crop species using the PlantVillage dataset, achieving over 99% accuracy. Ferentinos [9] further advanced this approach by applying multiple CNN architectures to various plant species, reinforcing the robustness of CNNs in agricultural image classification.

Too et al. [10] compared multiple architectures, including VGG16, Inception V4, and DenseNet121, and found DenseNet121 to outperform the others in both accuracy and computation time. Atila et al. [11] demonstrated the capabilities of EfficientNet-B4 and B5 in this domain, while Sutaji and Yıldız [12] proposed a combination of Vision Transformers (ViT) and MobileNetV2 to enhance model generalization.VGG16 has remained a popular architecture because of its consistent performance in transfer learning tasks. For instance, Chen et al. [13] used a modified VGG16 for maize and rice diseases, while Coulibaly et al. [14] applied transfer learning with VGG16 to identify mildew on pearl millet.

To improve performance and reduce complexity, studies have explored hybrid VGG-based models. Thomkaew and Intakosum [2](2022) integrated InceptionV3 modules into VGG16, increasing classification accuracy to 99.27%. Thakur and Gandhi [5] incorporated InceptionV7 with Global Average Pooling into a VGG16 backbone for tomato leaf disease detection. The role of hyperparameter optimization has gained prominence. Snoek et al. [6] introduced Bayesian Optimization as a method to improve neural network performance through intelligent hyperparameter tuning. Prabha & Chelliah [15] validated this approach in maize and corn disease detection using VGG16. Khan et al. [4] implemented a Bayesian-optimized multimodal deep hybrid model that achieved superior results in tomato disease classification. Likewise, Mustafa and Khan [16] integrated Bayesian Optimization into a hybrid CNN-RNN model for tomato leaf disease classification, further confirming the approach's efficacy.

Metaheuristic optimization algorithms such as Genetic Algorithms and Ant Colony Optimization have also been explored in this context. According to Abade et al. [17], these techniques can serve as complementary alternatives to Bayesian approaches in certain problem domains. However, very few studies have combined architectural enhancements with hyperparameter optimization within a single framework, especially for VGG16.

Research trends further indicate a growing interest in deploying lightweight CNN models for real-time diagnosis via mobile applications [18], integrating spatial attention [3], and using diverse datasets to enhance model robustness [19]. Despite this progress, relatively few studies have combined both architectural modifications and hyperparameter optimization within a unified VGG16-based framework focused on tomato leaf disease classification, an important gap addressed by this study.

To address this gap, the present research proposes a structurally improved VGG16 architecture integrated with Bayesian optimization for hyperparameter tuning, specifically focused on tomato leaf disease classification. The key contributions of this study are as follows:

1. The development of a lightweight VGG16-based model includes architectural enhancements like dropout, batch normalization, and reduced convolutional complexity to improve efficiency without sacrificing accuracy.
2. A systematic application of Bayesian optimization to identify optimal hyperparameter settings, including learning rate, input image size, dropout rate, and dense layer configuration.
3. Comprehensive experimental validation utilizing a publicly available tomato leaf disease dataset demonstrates the proposed model's superiority over the baseline VGG16 in terms of both accuracy and computational efficiency.

3 Research Methods

This section details the systematic approach adopted in optimizing the VGG16 architecture for the classification of tomato leaf diseases. The methodology is composed of three primary stages: dataset preprocessing, architectural modification of the VGG16 model, and hyperparameter optimization using Bayesian techniques.

3.1 Dataset and Preprocessing

The dataset used in this study is the Tomato Leaf Disease Detection dataset, which is publicly available on the Kaggle platform. It contains around 11,000 RGB images of tomato leaves, classified into ten categories: one representing healthy leaves and the other nine corresponding to specific disease types, such as bacterial spot, early blight, late blight, leaf mold, septoria leaf spot, spider mites, target spot, tomato mosaic virus, and yellow leaf curl virus.

To ensure balanced representation and enhance the generalization capability of the model, the dataset was randomly split into training (80%) and testing (20%) sets. Several preprocessing steps were performed as follows:

a. Resizing**:** All images were resized to fixed dimensions of 190×190 pixel.
b. Data Augmentation: Techniques such as horizontal flipping, random zoom, rotation, and shifting were applied to artificially expand the training set and mitigate overfitting.
c. Normalization: Pixel values were scaled to the [0,1] range by dividing all RGB channel values by 255.
d. Label Encoding**:** Class labels were one-hot encoded to fit the output format of the softmax activation in the final layer.

Data preprocessing pipeline illustration can be seen at Figure 1.

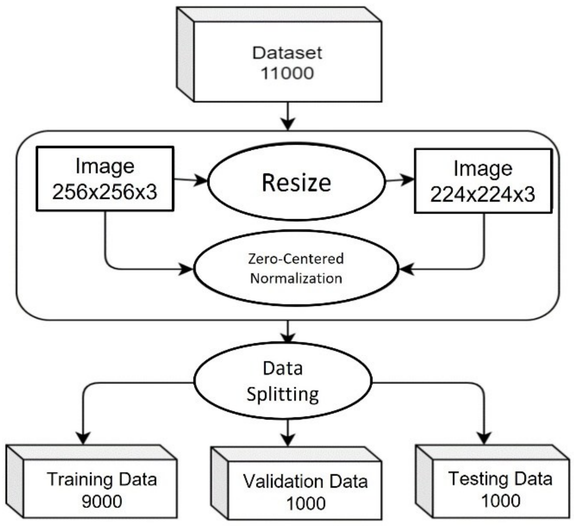

Figure 1 Data preprocessing pipeline illustration

3.2 Modified VGG16 Architecture

The original VGG16 architecture, while highly effective in generic image classification tasks, presents challenges for agricultural scenarios due to its rigid structure and extensive number of parameters. To address these limitations, this study proposes a modified version of the VGG16 model, designed to enhance learning efficiency and generalization while minimizing overfitting and computational overhead.

Key architectural changes include:

a. Convolutional Layers: The number of convolutional layers was reduced from 13 in the original model to only 6 in the modified version. This simplification reduces computational cost and avoids overfitting on domain-specific datasets.
b. Max Pooling Layers: Unlike the original VGG16 which applied max pooling after stacked convolutional blocks, the modified version introduces six max pooling layers, each applied after every convolutional layer, to progressively reduce spatial dimensions and computation.
c. Dropout Layers: Six dropout layers were implemented, each positioned after every max pooling layer to mitigate co-adaptation of neurons and enhance generalization. These were not present in the original architecture.
d. Batch Normalization: Six batch normalization layers were added directly after each dropout layer. These layers stabilize and accelerate the training process by normalizing the intermediate activations, enabling better convergence.
e. Dense Layers: The fully connected layers were retained as in the original architecture (three layers in total, including the softmax output), ensuring compatibility with the classification task of 10 tomato disease classes.
f. Kernel and Activation Settings: Across all convolutional layers, 3×3 kernels and stride size of 1 were maintained. ReLU was used as the activation function for all convolutional and dense layers, consistent with the original configuration. Zero-padding was not applied, leading to progressive reduction of image dimensions.
g. Input Size: The image input size was not fixed to 224×224 as in the original, but made tunable and optimized using Bayesian Optimization. This flexibility allows the model to adapt better to the specific characteristics of the tomato leaf dataset.
h. Output Layer: Both the original and modified models use a softmax activation function in the final dense layer to classify into 10 disease categories. The reference to 1000-unit output in some illustrations pertains to a generalized VGG16 version trained on ImageNet and not relevant to the target task in this study.

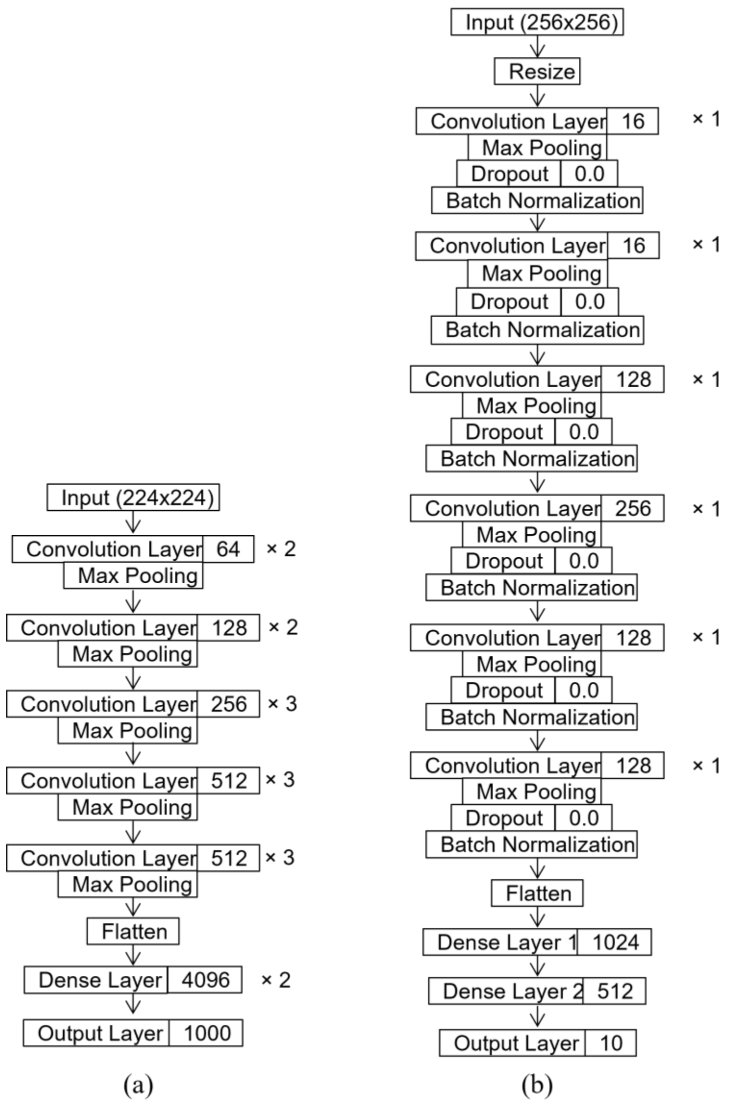

Figure 1 Ilustration of VGG16 Architecture (a) original (b) modified

Figure 1 illustrates the differences in architecture between the original VGG16 model and the modified VGG16 model. This modified architecture was developed alongside a Bayesian Optimization strategy that systematically searched for the optimal hyperparameter configuration, including the number of filters, dropout rates, and dense layer widths using Keras Tuner. The optimized parameter set was then used to train the modified model and benchmark its performance against the original VGG16.

3.3 Bayesian Optimization for Hyperparameter Tuning

Hyperparameter tuning plays a critical role in determining the performance and generalization ability of deep learning models. In this study, Bayesian Optimization was employed to systematically search for the optimal combination of hyperparameters for the modified VGG16 model during the training process. Unlike traditional search strategies such as grid search or random search, Bayesian Optimization uses a probabilistic surrogate model to model the objective function and an acquisition function to determine the most promising set of hyperparameters to evaluate next.

The goal of hyperparameter tuning in this research was to minimize validation loss while maximizing classification accuracy. The optimized hyperparameters include the learning rate, batch size, dropout rate, and the number of units in the fully connected layers. The search space for each hyperparameter was defined as follows:

a. Learning rate: 1×10^{-5} to 1×10^{-2} (log scale)
b. Dropout rate: 0.1 to 0.5
c. Dense layer size: 128 to 1024 units
d. Batch size: 16 to 64

Bayesian Optimization constructs a posterior distribution over the objective function and uses it to select the most promising hyperparameters to evaluate, based on the Expected Improvement (EI) acquisition function. The optimization process can be expressed as:

$$x^* = \arg max_{x \in X} \; \alpha(x; D_{1:t}) \tag{1}$$

Where $\alpha(x; D_{1:t})$ is the acquisition function conditioned on the data $D_{1:t} = \{(x_i, y_i)\}_{i=1}^{t}$, and x represents a point in the hyperparameter space X. The objective function *f(x)* models the validation performance (e.g., loss) associated with a particular hyperparameter configuration *x*.

The process was repeated for 50 trials, after which the best configuration was selected and used in the final training of the optimized VGG16 model. Table 1 shows the hyperparameter search space and optimal configuration, while the visualization of the most optimal modified VGG16 architecture with the best combination of hyperparameter values is shown in Figure 2. All experiments were conducted using a workstation equipped with an Intel Core i7-12700K CPU, 32GB RAM, and an NVIDIA RTX 3060 GPU with 12GB VRAM, running on Ubuntu 22.04 and TensorFlow 2.12.

Table 1. Hyperparameter Search Space and Optimal Configuration

Hyperparameter	Search Space	Optimal Value
Image Size	190–240 (stride=1)	190
Conv-1 Filters	{16, 32}	16
Conv-2 Filters	{16, 32, 64}	16
Conv-3 Filters	{16, 32, 64, 128}	128
Conv-4 Filters	{16, 32, 64, 128, 256}	256
Conv-5 Filters	{16, 32, 64, 128, 256, 512}	128
Conv-6 Filters	{16, 32, 64, 128, 256, 512, 1024}	128
Dropout-1	0.0–0.9 (stride=0.01)	0.0
Dropout-2	0.0–0.9 (stride=0.01)	0.0
Dropout-3	0.0–0.9 (stride=0.01)	0.0
Dropout-4	0.0–0.9 (stride=0.01)	0.0
Dropout-5	0.0–0.9 (stride=0.01)	0.4
Dropout-6	0.0–0.9 (stride=0.01)	0.0
Dense-1 Units	{16, 32, ..., 1024}	1024
Dense-2 Units	{16, 32, ..., 1024}	512

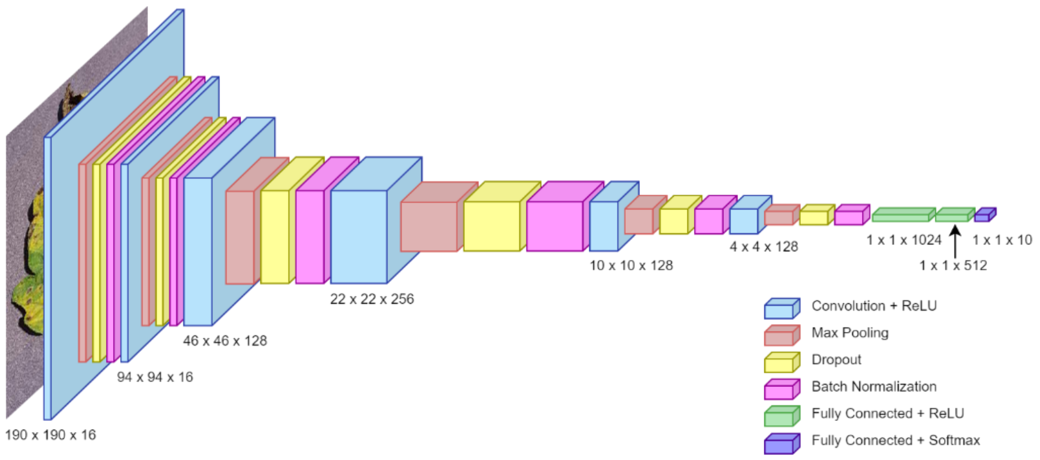

Figure 2 Visualization of the Modified VGG16 Architecture with the Best Combination of Hyperparameter Values

4 Results and Discussion

This section discusses the experimental results of the original and modified VGG16 models for tomato leaf disease classification. Performance metrics such as training/validation accuracy and loss, test accuracy, and confusion matrix analysis were used to evaluate the models. Additionally, the benefits of using Bayesian Optimization for hyperparameter tuning are explained, followed by a reflection on future enhancements.

4.1 Evaluation of Model Performance

The evaluation of model performance was conducted through a series of experiments comparing the original and modified versions of the VGG16 architecture. The models were assessed based on their training and validation behavior, test accuracy, loss, and class-wise predictions using the confusion matrix. The original VGG16 model was implemented without additional regularization or architectural tuning. It achieved a high training accuracy of 99.99%. However, this performance was not sustained during validation. Its validation accuracy dropped to 93.8%, and the validation loss rose sharply to 30.62%, clearly indicating overfitting. The model memorized training samples but failed to generalize on unseen data.

On the other hand, the modified VGG16 model was improved with architectural enhancements such as dropout layers, batch normalization, and Bayesian-tuned dense layers. As a result, it achieved a more balanced performance in training and validation. It reached a training accuracy of 99.83% and a validation accuracy of 98.3%. The validation loss was significantly lower at 4.86%, highlighting a greatly improved generalization capability. These enhancements demonstrate the effectiveness of using regularization and structural tuning to adapt CNNs for relatively small, domain-specific datasets like tomato leaf disease images. A comparison of the training and validation score history between the original VGG16 and modified VGG16 models is shown in Figure 3.

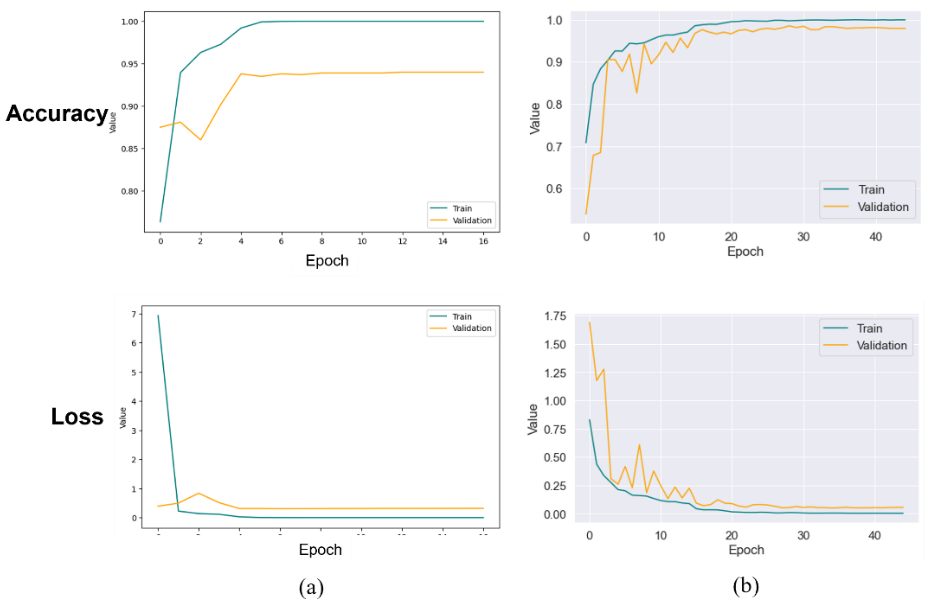

Figure 3 Train and Validation Score History on VGG16 Model (a) original (b) modified

The benefits were further validated on the test dataset. The original model achieved a test accuracy of 89.0% with a loss of 65.83%, while the optimized model attained an impressive accuracy of 97.1% and a significantly reduced test loss of 10.66%. This gap underscores the importance of systematic optimization in enhancing predictive robustness. A comparison of the test scores between the original VGG16 and modified VGG16 models is shown in Figure 4.

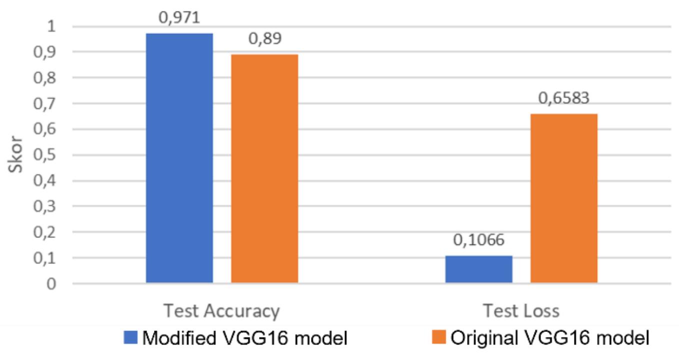

Figure 4. Comparison of Test Score

To complement this global performance overview, a detailed class-wise analysis was performed using the confusion matrix. The following insights were observed:

- Tomato Bacterial Spot (Class 0) was classified almost perfectly in both models, with nearly no confusion, indicating distinct features easily detected by the CNN.
- Early Blight (Class 1) and Late Blight (Class 2) were frequently confused in the original model. The optimized model significantly reduced this overlap, although minor misclassifications remained due to their similar visual patterns.
- Leaf Mold (Class 3) and Septoria Leaf Spot (Class 4) initially posed classification difficulties. Post-optimization, the model learned better feature representations, leading to improved accuracy and reduced misclassification with Target Spot (Class 6).

- Two-Spotted Spider Mites (Class 5) showed major improvement. The model initially struggled to recognize pest damage patterns, but after optimization, accuracy increased markedly, indicating better sensitivity to subtle texture disruptions.
- Target Spot (Class 6) was the most challenging class. It was frequently misclassified as Septoria or Leaf Mold. The optimized model improved performance but some confusion persisted, suggesting that further enhancement may be needed.
- Yellow Leaf Curl Virus (Class 7) and Tomato Mosaic Virus (Class 8) were classified with high accuracy due to their distinctive patterns (e.g., leaf curling, mottling), with the latter achieving nearly perfect results.
- Healthy leaves (Class 9) were occasionally confused with early disease symptoms, but the optimized model successfully distinguished them, reducing false positives to a minimum.

Figure 5 illustrates the side-by-side confusion matrices for both models, while Table 2 provides the corresponding label encoding. Overall, these results validate the effectiveness of the architectural refinements and Bayesian hyperparameter optimization strategy. They not only improved general accuracy but also enhanced the model's class-wise discrimination, particularly for diseases with overlapping visual symptoms.

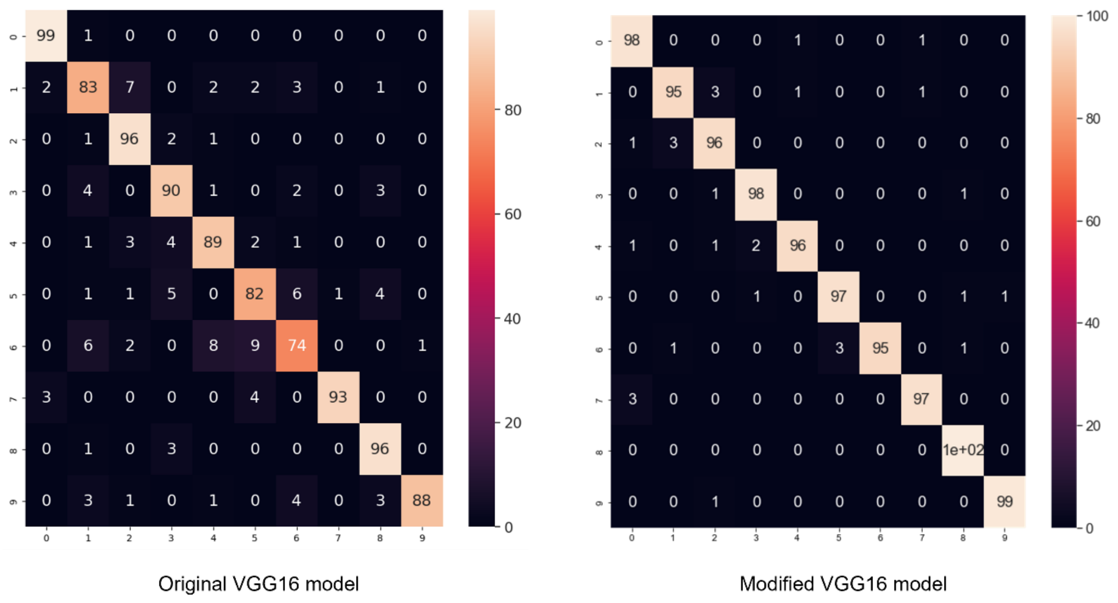

Figure 5. Comparison of Test Confusion Matrix

Table 2. Label Encoding

Label	Class
0	Tomato Bacterial Spot
1	Tomato Early Blight
2	Tomato Late Blight
3	Tomato Leaf Mold
4	Tomato Septoria Leaf Spot
5	Tomato Two-Spotted Spider Mites
6	Tomato Target Spot
7	Tomato Yellow Leaf Curl Virus
8	Tomato Mosaic Virus
9	Tomato Healthy

4.2 Impact of Bayesian Hyperparameter Optimization

Bayesian Optimization significantly contributed to the final model's performance by effectively identifying the best hyperparameter configurations. The optimal setup included an input size of 190×190 pixels, a dropout rate of 0.3, and fully connected layers with 1024 and 512 units. This configuration provided a balance between model capacity and regularization. Compared to manual tuning or exhaustive grid search, Bayesian Optimization reduced the number of training iterations needed to converge while maintaining high accuracy. Its strategic exploration of the hyperparameter space resulted in faster and more effective training, which is especially valuable in deep learning tasks with costly evaluations functions.

4.3 Limitations and Future Work

Despite its success, the model still exhibited occasional confusion between visually similar diseases, such as Early Blight and Late Blight. This indicates a need for more discriminative feature extraction or attention mechanisms. In future work, integrating attention layers or contextual metadata may further enhance classification performance. Additionally, deploying and testing the model in real-world conditions is essential to validate its robustness beyond controlled datasets. Future research may also explore integrating ensemble learning or multi-modal inputs to improve diagnostic accuracy further. Furthermore, this study did not explicitly measure computational performance, which may be important for real-time or resource-constrained applications. Future work should include a more comprehensive computational evaluation to assess deployment feasibility in embedded systems or mobile devices.

5 Conclusion

This research presents a robust and efficient deep learning framework for tomato leaf disease classification by optimizing the VGG16 architecture through Bayesian hyperparameter tuning. The proposed modifications, including architectural simplification, the integration of dropout and batch normalization, and data-driven tuning of hyperparameters, led to significant performance improvements over the baseline VGG16 model. By systematically evaluating multiple configurations, the modified VGG16 model achieved a test accuracy of 97.1% and reduced test loss to 10.66%, outperforming the original model in both accuracy and generalization capability. Utilizing Bayesian Optimization enabled the efficient exploration of a complex hyperparameter space, yielding configurations that balanced model complexity and predictive power. Additionally, analyzing the confusion matrices provided insights into the interpretability and robustness of the model. In conclusion, this study validates the synergy of architectural simplification and Bayesian Optimization in designing deep learning models for agricultural disease classification. It opens avenues for deploying reliable and efficient AI solutions in smart farming systems, contributing meaningfully to global efforts in sustainable agriculture and food security.

Acknowledgement

Thank you to the team at the Intelligent Systems Laboratory and Visual Computing Laboratory at Diponegoro University for their considerable assistance in carrying out modeling and testing. This manuscript benefited from language refinement using OpenAI ChatGPT (GPT-4), whose outputs were reviewed and edited by the authors to maintain academic integrity and clarity accuracy.

Bibliography

[1] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” *3rd International Conference on Learning Representations* (ICLR 2015), Computational and Biological Learning Society, pp. 1–14, 2015. https://doi.org/10.48550/arXiv.1409.155

[2] J. Thomkaew and S. Intakosum, “Modified VGG16 with InceptionV3 Modules for Plant Disease Classification”. *Procedia Computer Science*, 167, pp. 302–309, 2022. https://doi.org/10.1016/j.procs.2020.03.238

[3] M. O. Faruk, M. R. Islam, & M. T. Hasan, “A Deep Learning Model Using Attention Mechanism For Plant Disease Classification”. *Computers and Electronics in Agriculture*, vol. 205, 107646, 2023. https://doi.org/10.1016/j.compag.2023.107646

[4] M. A. Khan, A. U. Rehman, and A. Mustafa, “Optimized Deep Multimodal Tomato Leaf Disease Classification using Bayesian-Tuned CNN. Applied *Sciences*, vol. 14, No. 2, 1862, 2024. https://doi.org/10.3390/app14021862

[5] A. Thakur and T. K. Gandhi, “Improvement In Classification Approach in Tomato Leaf Disease Detection using Modified Deep CNN”. *Procedia Computer Science*, vol. 167, pp. 293–301, 2022. https://doi.org/10.1016/j.procs.2020.03.237

[6] J. Snoek, H. Larochelle, and R. P. Adams, Practical Bayesian Optimization of Machine Learning Algorithms. In Advances in Neural Information Processing Systems, Vol. 25. 2012. https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf

[7] B. Khan, S. Das, Fahim, N.S. et al. “Bayesian Optimized Multimodal Deep Hybrid Learning Approach for Tomato Leaf Disease Classification”. *Sci Rep 14*, 21525, 2024. https://doi.org/10.1038/s41598-024-72237-x

[8] S. P. Mohanty, D. P. Hughes, & M. Salathé, “Using Deep Learning for Image-Based Plant Disease Detection”. *Frontiers in Plant Science*, vol. 7, 1419, 2016. https://doi.org/10.3389/fpls.2016.01419

[9] K. P. Ferentinos, “Deep Learning Models for Plant Disease Detectiona nd Diagnosis”. *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018. https://doi.org/10.1016/j.compag.2018.01.009

[10] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, “A Comparative Study of Fine-Tuning Deep Learning Models for Plant Disease Identification”. *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019. https://doi.org/10.1016/j.compag.2018.03.032

[11] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, “Plant leaf disease classification using EfficientNet deep learning model”. *Ecological Informatics*, vol. 61, 101182, 2021. https://doi.org/10.1016/j.ecoinf.2020.101182

[12] Y. Sutaji and O. Yıldız, “Vision Transformer Meets Convolutional Neural Network for Plant Disease Classification”. *Ecological Informatics*, 69, 101678, 2022. https://doi.org/10.1016/j.ecoinf.2022.101678

[13] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, “Using Deep Transfer Learning for Image-Based Plant Disease Identification”. *Computers and Electronics in Agriculture*, 173, 105393, 2020. https://doi.org/10.1016/j.compag.2020.105393

[14] S. Coulibaly, B. Kamsu-Foguem, and D. Kamissoko, “Deep Learning for Plant Diseases: Detection and Diagnosis”. *Computers and Electronics in Agriculture*, 182, 105993, 2021. https://doi.org/10.1016/j.compag.2021.105993

[15] R. Prabha and P. R. Chelliah, “Hyperparameter Optimization for Transfer Learning of VGG16 for Disease Identification in Corn Leaves using Bayesian Optimization”. *Computers and Electronics in Agriculture*, 187, 106290, 2022. https://doi.org/10.1016/j.compag.2021.106290

[16] A. Mustafa and M. A. Khan, “Optimizing Plant Disease Classification with Hybrid Convolutional Neural Network–Recurrent Neural Network and Liquid Time-Constant Network”. *Applied Sciences*, 12(19), pp. 9118, 2022. https://doi.org/10.3390/app12199118

[17] A. S. Abade, P. A. Ferreira, and F. B. Vidal, “Metaheuristic Methods for Hyperparameter Tuning in Deep Learning and Their Application in Plant Leaf Disease Detection: A Survey”. *Artificial Intelligence in Agriculture*, 6, pp.1–18, 2022. https://doi.org/10.1016/j.aiia.2022.05.001

[18] A. K. Rangarajan and R. Purushothaman, “Tomato Crop Disease Classification using Pre-Trained Deep Learning Algorithm”. *Procedia Computer Science*, 180, pp. 586–593, 2021. https://doi.org/10.1016/j.procs.2021.01.311

[19] S. T. Y. Ramadan, T. Sakib, F Al Farid, Md.S. Islam, J.B. Abdullah, and Md. R. Bhuiyan, "Improving Wheat Leaf Disease Classification: Evaluating Augmentation Strategies and CNN-Based Models With Limited Dataset," in *IEEE Access*, vol. 12, pp. 69853-69874, 2024, doi: 10.1109/ACCESS.2024.3397570.