

Fine-Tuned Transfer Learning with InceptionV3 for Automated Detection of Grapevine Leaf Diseases

Miftahus Sholihin^{*1}, Moh. Rosidi Zamroni², Lilik Anifah³, Mohd Farhan Md Fudzee⁴, Mohd Norasri Ismail⁵

^{1,2}Informatic Engineering, Universitas Islam Lamongan, Indonesia

³Electrical Engineering Department, Universitas Negeri Surabaya, Indonesia

^{4,5}Faculty of Computer Sciences and Information Technology, University Tun Hussein Onn Malaysia, Malaysia

Email: miftahus.sholihin@unisla.ac.id

Received : May 13, 2025; Revised : Jun 24, 2025; Accepted : Aug 16, 2025; Published : Oct 21, 2025

Abstract

Grape leaf diseases pose a major threat to vineyard productivity, making early and accurate detection essential for modern grape plantation management. Despite advancements in computer vision, challenges remain in differentiating diseases with visually similar symptoms. This study addresses that gap by developing a grape leaf disease classification system using a fine-tuned deep learning model based on the InceptionV3 architecture. Three training scenarios were conducted with fixed parameters batch size of 32 and learning rate of 0.001 while varying the number of epochs (25, 50, and 75). Results showed a consistent improvement in classification accuracy with increased training epochs, reaching 98.64%, 98.78%, and 99.09% respectively. Confusion matrix analysis revealed that most misclassifications occurred between visually similar diseases such as Black Rot and ESCA, but error rates declined as the number of epochs increased. Rather than merely applying transfer learning, this research highlights the impact of systematic tuning specifically epoch count optimization in enhancing model accuracy for difficult to distinguish disease classes. These findings underscore the urgency of developing high performance, automated disease detection tools to support precision agriculture and sustainable crop health monitoring.

Keywords : Agriculture, Deep learning, Grape leaf disease, InceptionV3, Transfer learning.

This work is an open access article and licensed under a Creative Commons Attribution-Non Commercial 4.0 International License



1. INTRODUCTION

Grapes are fruit plants with high economic value and are widely cultivated worldwide, especially in the Mediterranean region. The global grape industry has a tremendous market value, exceeding 189 billion USD [1]. Grapes are not only consumed as fresh fruit, but are also processed into various products such as drinks and sweet foods. However, like other plants, grapes are susceptible to multiple diseases that require immediate treatment.

Diseases in grape plants can cause significant economic losses [2]. Studies have shown that grape diseases can result in yield losses of between 10% and 30% [3]. In the worst cases, losses due to disease can even reach 80% of the total harvest [1]. Some common diseases that attack grape plants include black rot, ESCA, leaf blight, powdery mildew, and downy mildew [4].

Rapid and accurate detection and diagnosis of grapevine diseases are essential for effective vineyard management [5]. Traditional grapevine disease identification methods generally rely on human visual observation [6], [7]. This approach has several drawbacks. It is time-consuming, requires expert personnel, and is prone to subjective errors [8]. Consequently, delays in disease identification and control measures often occur. These limitations drive the urgent need for automated systems capable of rapid and accurate detection and classification of grapevine leaf diseases.

In recent years, deep learning methods, especially Convolutional Neural Networks (CNN), have been widely applied in agriculture [9], [10], [11], [12], [13], [14]. Various variants of CNN-based methods have been developed to identify and monitor plant diseases [15]. The adaptation of CNN models for specific agricultural problems remains an active and innovative area of research. Ji et al [2] and Liu et al [15] specifically focused on grape leaf disease, achieving accuracies of 98.57% and 97.22%, respectively, by modifying the basic CNN architecture. Significant progress was also seen in other commodities. For instance, Zhang et al [16] adapted AlexNet for cucumber disease identification (94.65%), while Thet et al [17] modified VGG16 to achieve 98.4% accuracy for five leaf disease classes. These findings strengthen the hypothesis that transfer learning from pre-trained CNN models can overcome data limitations [18]. Recent studies have further demonstrated the potential of fine-tuned transfer learning models such as InceptionV3 and EfficientNetB0 in classifying grapevine leaf diseases with high accuracy (92%–99.02 %) [8], [19], [20], [21]. These models have enabled early detection and improved disease management strategies, further validating their role in modern agriculture.

On the other hand, the InceptionV3 architecture introduced by Szegedy et al [22] is one of the CNN models in the GoogLeNet family that offers a balance between network depth, computational efficiency, and classification performance. InceptionV3 has been proven effective in various everyday image classification tasks and has been widely used in transfer learning applications in multiple domains [23], [24], [25], including medical and agricultural fields [26], [27], [28], [29]. The transfer learning strategy allows pre-trained weights from large datasets (such as ImageNet) to speed up the training process and improve accuracy on smaller, specific datasets.

Although the transfer learning approach with CNN has been widely applied to plant disease classification, studies that specifically apply fine-tuning to InceptionV3 for grape leaf disease are still relatively limited. Most studies only use pre-trained models as feature extractors without further adjustments to the model parameters. In fact, fine-tuning the upper layers of the network enables the model to adapt to the target dataset more effectively. This adjustment can improve both accuracy and generalization performance. However, limitations remain in model generalization to varied datasets and environmental conditions, as highlighted by Das et al. [30] and Tamilmani and Khan [31]. Computational cost also remains a concern when scaling to large datasets. These practical concerns highlight the need for efficient and robust training strategies.

Based on the literature review, a research gap has been identified in applying deep learning techniques, particularly advanced architectures such as InceptionV3, for detecting and classifying grape leaf diseases. To address this gap, the present study proposes a novel contribution by implementing a transfer learning approach based on the InceptionV3 architecture, optimized through fine-tuning techniques to improve classification accuracy. InceptionV3 was selected due to its ability to extract multiscale features efficiently through its modular design. This research is expected to broaden the application of InceptionV3 in plant disease classification and offer a more accurate and computationally efficient solution for grape leaf disease detection. To the best of our knowledge, this study is the first to systematically investigate the impact of fine-tuning epoch variations on the performance of InceptionV3 for grape leaf disease classification. This research is expected to broaden the application of InceptionV3 in plant disease classification. It also offers a more accurate and computationally efficient solution for grape leaf disease detection. Furthermore, this work contributes to the conversation on responsible AI in agriculture by acknowledging ethical considerations such as data bias, privacy, and the impact on traditional farming practices [32], [33]. Ultimately, the findings are expected to support enhanced productivity and sustainability in the grape plantation industry.

2. METHOD

This research was conducted through four main stages, as shown in Figure 1. The first stage is data collection, where the dataset used was obtained from the Kaggle platform. The dataset contains a collection of grape leaf images categorized into several classes.

The next stage is data pre-processing. At this stage, a series of processes are carried out to ensure that the data is in optimal condition before being used in model training. Pre-processing activities include normalizing pixel values into a scale of $[0,1]$ to speed up the convergence process and data augmentation in the form of rotation, flipping, and zooming to increase the diversity of training data and prevent overfitting. This process aims to strengthen the model's generalization to image variations in the real world.

The third stage is the utilization of pre-trained models. In this context, the InceptionV3 architecture, which has been pre-trained on a large-scale dataset such as ImageNet, is used. The transfer learning approach allows the utilization of visual features learned by the model, so that training on a specific dataset (grape leaf images) can be done more efficiently and accurately, even with limited data. The pre-trained model is selected and adjusted to the dataset's characteristics, and adjustments to the output layer according to the number of target classes.

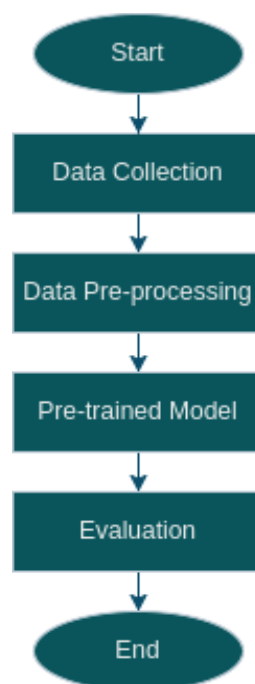


Figure 1. Research Stage

The final stage is the model performance evaluation, which is carried out using metrics such as accuracy, precision, recall, and F1-score, which are extracted from the confusion matrix. This evaluation assesses the model's generalization ability to test data and detect potential bias or class imbalance. The evaluation results are the basis for determining the approach's effectiveness and the potential for further development.

2.1. Data Collection

The initial stage in this study is data collection. In this study, the data used are images of grape leaves labeled according to their health conditions and the type of disease that attacks them. The dataset is supervised and consists of four classes: black rot, ESCA (black measles), leaf blight, and healthy

leaves. This dataset was obtained from the Kaggle site via the link (<https://www.kaggle.com/datasets/rm1000/grape-disease-dataset-original>) uploaded by Mandal and Gouda [34]. This dataset was chosen because it has good image quality and has been widely used in plant disease classification research. The number of images in each class is presented in the following Table 1.

Table 1. Grape Disease Dataset.

Class Name	Training	Testing	Total
Black Rot	1888	472	2360
ESCE	1920	480	2409
Leaf Blight	1722	430	2152
Healthy	1692	423	2115

To provide a visual representation of the data used, Figure 2 are sample images from each class.



Figure 2. Sample of Grape Leaf.

2.2. Data Pre-processing

After all the grape leaf image data has been successfully collected, the next stage is data pre-processing to ensure that the data is in optimal condition for use by the deep learning model. This stage begins with the resizing process, which is changing all images to a uniform size, for example, 224×224 pixels, to match the standard input of the pre-trained model architecture. Furthermore, the image pixel values are normalized into the $[0,1]$ range by dividing the RGB values by 255. This step is essential to speed up the convergence process during model training and maintain learning stability.

To increase data variability and enhance the model's generalization capability, several data augmentation techniques are applied, including random rotation, horizontal flipping, zoom, and brightness adjustment. These augmentations are applied only to the training subset to minimize the risk of overfitting, especially when the dataset size is relatively small.

The dataset is divided into three main subsets: training, validation, and testing. The training and validation subsets are generated using stratified splitting, implemented through the ImageDataGenerator class from the Keras library. A validation split of 10% (`validation_split=0.1`) is used to ensure that the class distribution remains balanced between the training and validation sets. The subsets are created internally using the `flow_from_directory()` method by specifying the `subset='training'` and `subset='validation'` parameters. Meanwhile, the test set is stored in a separate directory and loaded using a dedicated ImageDataGenerator instance with only rescaling (no augmentation applied). Additionally, `shuffle=False` is used to preserve the order of samples, which is important for accurate performance evaluation and comparison with ground truth labels.

2.3. Pre-trained Model

This study uses a transfer learning approach using the InceptionV3 architecture pre-trained on the ImageNet dataset. The base model is initialized with the `include_top=False` parameter to remove the default classification part and retain the convolution layer as a feature extractor. All layers in the base model are frozen so that the learned weights are not updated during the training process, so that the general features that have been discovered are maintained.

A unique classification head has been added to adapt the model to the targeted classification task. First, a 2D Global Average Pooling layer is applied to reduce the spatial dimension of the output features to a one-dimensional vector of size 2048. Dropout follows this layer with a deletion rate 0.5 to reduce the risk of overfitting by randomly deactivating some neurons during training.

Next, a Dense layer with 256 units and a ReLU activation function is added to build a more specific feature representation for the classification task. Finally, the output layer is a Dense layer with four units, and a softmax activation function is used to generate probability predictions for the four target classes.

During the fine-tuning phase, several hyperparameters were optimized, including setting the initial learning rate to 0.001 using the Adam optimizer. A learning rate scheduler was implemented using `ReduceLROnPlateau` to reduce the learning rate when the validation loss plateaued. Early stopping with a patience of 10 epochs was also employed to prevent overfitting. The model was trained with a batch size of 32 and a maximum of 75 epochs. Callbacks such as `ModelCheckpoint` were used to save the best-performing model based on validation accuracy. The complete structure of the customized InceptionV3 model architecture is shown in Figure 3, which illustrates the image processing flow from input to generating class predictions.

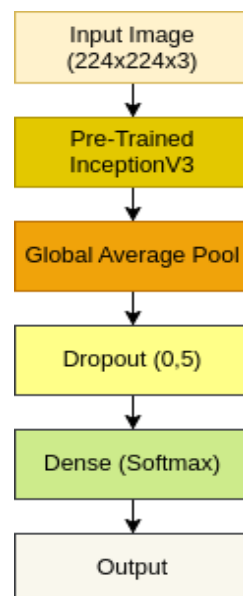


Figure 3. Proposed Method.

2.4. Evaluation

After the model training process is completed, the next step is to evaluate the model's performance using test data that has never been involved in the training or validation process. This evaluation aims to measure the model's generalization ability to new data and identify potential weaknesses in classification performance. Several evaluation metrics used in this study include accuracy, representing the proportion of correct predictions compared to all projections made.

In addition, precision, recall, and F1-score are also calculated using the following mathematical Equations (1), (2), (3), and (4).

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$Accuracy = \frac{TP+FP}{TP+FP+TN+FN} \quad (3)$$

$$F1 - score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

Where TP = True Positive, FP = False Positive, and FN = False Negative.

Precision measures how accurate the optimistic predictions produced by the model are, while recall describes how much the model can capture all actual positive samples. The F1-score, being the harmonic mean of precision and recall, provides a more balanced picture of model performance, particularly in the presence of class imbalance.

In addition to these metrics, a confusion matrix is also used to visualize the distribution of predictions between classes, including the number of correct predictions (true positives and true negatives) and misclassifications (false positives and false negatives). The confusion matrix is implemented using the `confusion_matrix` function from `sklearn.metrics`. Analysis of the confusion matrix provides more detailed insights into which classes are most often misclassified, so that it can be used as a basis for efforts to improve model performance in subsequent iterations.

3. RESULT

In this study, training and testing of image classification models were carried out using three scenarios based on the number of epochs, namely 25, 50, and 75. Other parameters used remained constant: the batch size of 32, the optimizer using Adam, and the learning rate of 0.001. Model performance was evaluated by analyzing the accuracy and loss graphs at the training stage and the confusion matrix analysis at the testing stage.

3.1. Training Process

The model training process was carried out with three different scenarios based on the number of epochs, namely 25, 50, and 75 epochs. The training parameters used for these three scenarios were a batch size of 32, an optimizer using Adam, and a learning rate of 0.001.

In the first scenario (25 epochs), the training accuracy showed a gradual increase from the initial epoch and approached 95% by the final epoch, while the validation accuracy fluctuated around 97% as shown in Figure 4. The training and validation loss values also decreased overall, although the validation loss experienced slight fluctuations, indicating some instability in generalization. This suggests that the model began learning relevant features but had not yet reached optimal stability.

In the second scenario (epoch 50) shown in Figure 5, the graph shows a more consistent increase in training accuracy compared to the first scenario. Training accuracy is close to 96%, and validation accuracy is around 97%. The training and validation loss values also experience a more significant decrease compared to the previous scenario, indicating better model performance. This suggests the model has started to generalize better to unseen data and is capturing more relevant feature patterns, especially in complex disease classes.

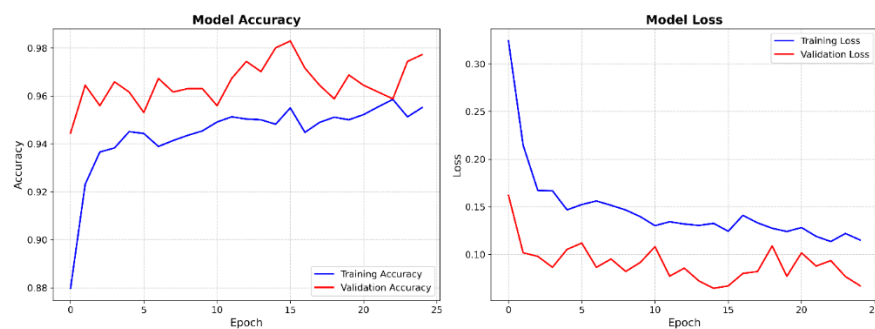


Figure 4. Training, Validation Accuracy, Training Loss, And Validation Loss With Epoch 25.

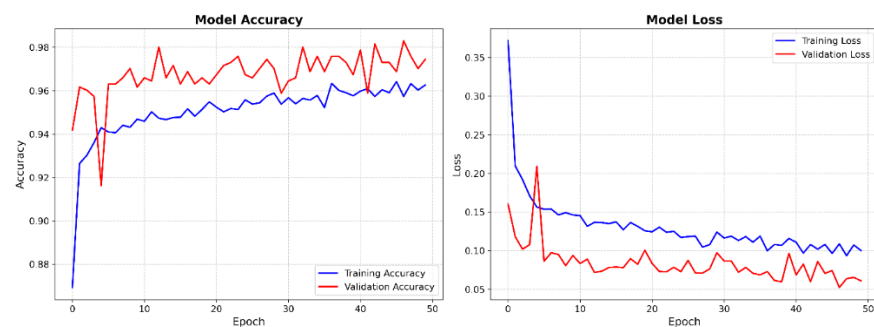


Figure 5. Training, Validation Accuracy, Training Loss, And Validation Loss With Epoch 50.

Meanwhile, Figure 6 is the third scenario (epoch 75, shown in Figure 6), the graph shows a more stable increase in training and validation accuracy. The training accuracy is above 96%, while the validation accuracy reaches a range of more than 97%. The loss values in training and validation experienced a sharper decline, with better stability compared to the previous scenario. This shows that the model has learned the data more effectively as the number of epoch increases. This trend implies that increasing epochs have enabled the model to extract more discriminative features, which are particularly useful in distinguishing visually similar classes.

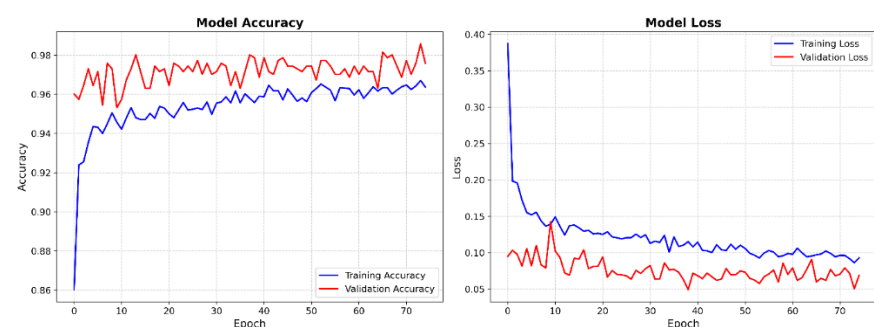


Figure 6. Training, Validation Accuracy, Training Loss, And Validation Loss With Epoch 75

3.2. Testing Process

This study evaluates the performance of the classification model using three different scenarios based on the number of epochs, namely 25, 50, and 75 epochs, with a batch size parameter of 32 and a learning rate of 0.001. Performance evaluation is carried out using a confusion matrix on the test data.

The model testing results at the 25 epoch configuration are illustrated by the confusion matrix in Figure 7. Overall, the model was able to classify most of the test data correctly, although notable misclassifications occurred in certain classes. The Black Rot class was accurately identified in 458 instances, with 14 samples misclassified as ESCA. Conversely, the ESCA class experienced a higher

rate of misclassification, with 32 samples incorrectly predicted as Black Rot, while 448 were correctly classified. The Leaf Blight class demonstrated excellent performance, achieving perfect classification with all 423 samples correctly predicted. The Healthy class also performed well, with 427 correct predictions and only 3 samples misclassified as Black Rot. These results suggest that at an early training stage (25 epochs), the model still struggles to differentiate between visually similar classes, particularly Black Rot and ESCA, which both exhibit necrotic spots and leaf discoloration patterns that may confuse the model's convolutional filters. This indicates that with a limited number of training cycles, the feature representations learned by the model are not yet sufficiently discriminative. In contrast, the strong classification performance in the Leaf Blight and Healthy classes implies that the distinguishing visual features of these categories are more easily captured even with shallow training. Overall, although the classification results at 25 epochs are promising, increasing the number of training epochs is necessary to reduce inter-class confusion and improve the model's generalization capability.

The model's performance in the testing phase with a training configuration of 50 epochs is depicted in the confusion matrix in Figure 8. Compared to the previous scenario with 25 epochs, the results show an overall improvement in classification accuracy and a reduction in misclassifications. For the Black Rot class, the model correctly classified 456 instances, with 16 samples misclassified as ESCA. Conversely, for the ESCA class, 454 samples were correctly identified, while 25 were misclassified as Black Rot, and 1 instance was incorrectly labeled as Healthy. The Leaf Blight class maintained its strong performance, achieving 100% accuracy with all 423 instances correctly classified, indicating that the features of this class continue to be reliably captured by the model. The Healthy class also exhibited near-perfect classification, with 428 correct predictions and only 1 sample misclassified as Black Rot, and 1 as ESCA.

These results suggest that increasing the number of training epochs enhances the model's ability to generalize and discriminate between visually similar classes. The notable reduction in misclassifications between Black Rot and ESCA although still present indicates the model is learning more refined and discriminative features with extended training. Additionally, the model's robustness across Leaf Blight and Healthy classes demonstrates its consistency in recognizing distinct visual characteristics. Overall, the 50 epoch configuration reflects a positive progression in classification performance, reducing inter-class confusion and further strengthening the model's reliability on unseen data.

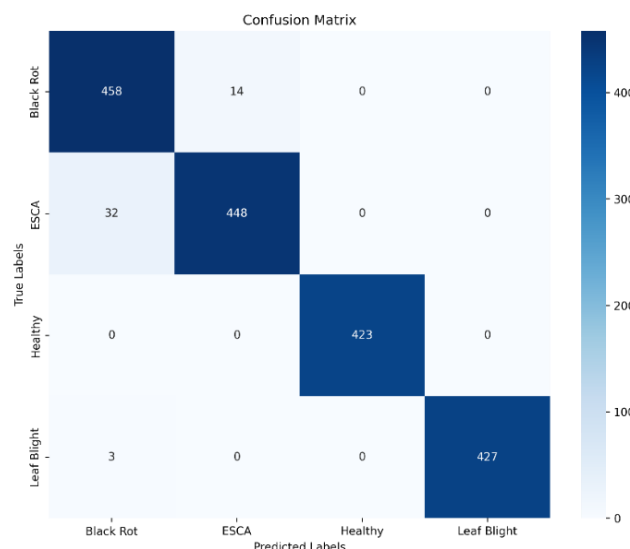


Figure 7. Confusion Matrix with Epoch 25.

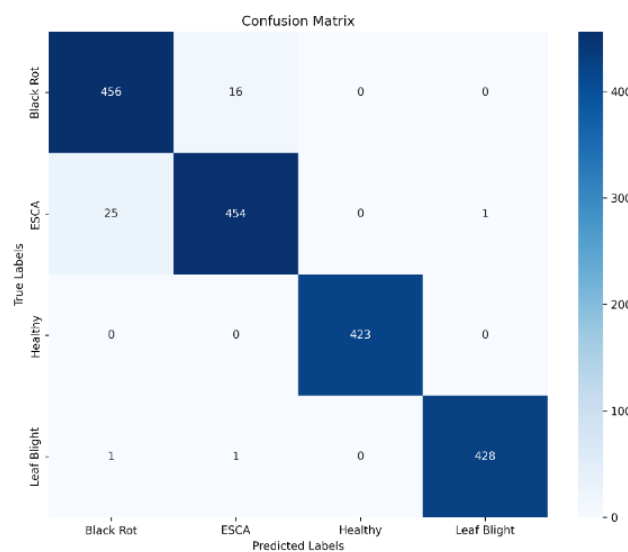


Figure 8. Confusion Matrix with Epoch 50.

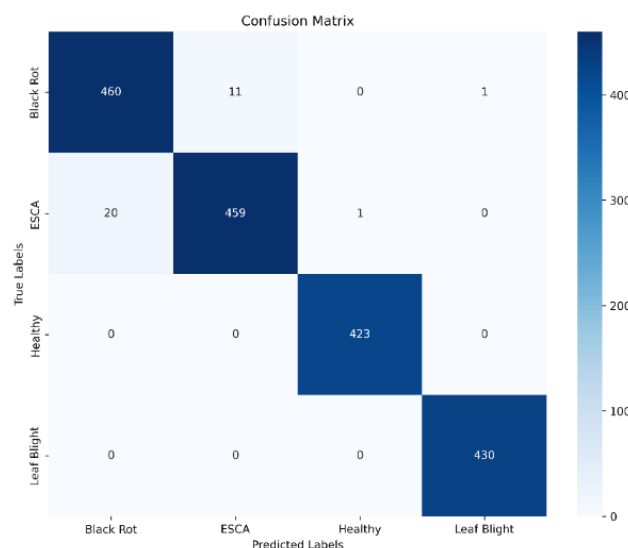


Figure 9. Confusion Matrix with Epoch 75.

The confusion matrix for the model trained with 75 epochs, shown in Figure 9, indicates the best overall performance among all tested configurations. The classification results show a notable reduction in misclassifications, particularly between the visually similar Black Rot and ESCA classes. For the Black Rot class, the model correctly identified 460 samples, with only 11 misclassified as ESCA and 1 as Healthy. Conversely, the ESCA class achieved 459 correct predictions, with 20 misclassified as Black Rot and 1 instance incorrectly identified as Leaf Blight.

The Leaf Blight class maintained perfect classification performance, with all 423 samples correctly predicted and zero errors, as in the previous scenarios. The Healthy class also achieved flawless classification, with 430 correct predictions and no misclassifications representing an improvement over earlier epochs.

These results demonstrate that increasing the number of epochs to 75 significantly improves the model's learning stability and feature discrimination, particularly in distinguishing between challenging class pairs such as Black Rot and ESCA. The continued accuracy of the Leaf Blight and Healthy classes further highlights the robustness and generalization capacity of the model. The overall distribution of correct predictions, along with the minimal misclassification rate, reinforces the finding that 75 epochs

represent the optimal training duration in this study, offering a well-balanced trade-off between accuracy, generalization, and computational efficiency.

To provide a clear comparison of model performance across different training configurations, an aggregated metric table is presented in Table 2.

From the results in Table 2, it can be concluded that increased epochs contribute significantly to better classification performance. In particular, 75 epochs provided the highest and most consistent accuracy, precision, recall, and F1-score.

Table 2. Overall Result Of Proposed Method.

Epoch	Precision	Recall	Accuracy	F1-score
25	97,47	97,42	98,64	97,42
50	97,56	97,56	98,78	97,56
75	98,26	98,27	99,09	98,26

4. DISCUSSIONS

The results of the model performance evaluation on three configurations of the number of epochs 25, 50, and 75 show a trend of increasing performance as the number of training cycles increases, as presented in Table 2. All evaluation metrics, namely precision, recall, accuracy, and F1-score, show consistent improvements with the best results achieved at the 75th epoch.

At the 25th epoch, the model recorded a precision of 97.47%, a recall of 97.42%, an accuracy of 98.64%, and an F1-score of 97.42%. These metrics indicate that in the early stages of training, the model can classify with a relatively low error rate. When the number of epochs was increased to 50, there was a slight but consistent improvement in all metrics, namely, precision and recall increased to 97.56%, while accuracy increased to 98.78%. This shows that adding training cycles strengthens the model's generalization ability.

Optimal performance was achieved at epoch 75, with precision and recall values of 98.26% and 98.27%, respectively, accuracy of 99.09%, and F1-score of 98.26%. These values reflect the stability and depth of the model's understanding of data patterns without indicating significant overfitting. The balance between precision and recall values also shows that the model cannot only minimize positive prediction errors, but is also quite sensitive in detecting all instances of the target class. This is reinforced by the high F1-score value, which indicates that the trade-off between precision and recall has been achieved optimally.

Overall, the gradual improvement in all validation metrics proves that the longer training process improves accuracy and consistency of model performance. Considering the aspects of accuracy, stability, and efficiency, it can be concluded that the training configuration with 75 epochs is the most optimal parameter to be applied in this study.

The model's performance in this study shows significant advantages compared to previous approaches in digital image-based leaf disease classification. Previous studies have proposed various strategies to improve classification accuracy, but with varying methods and complexities.

In recent years, a wide range of deep learning approaches have been explored for grapevine leaf disease classification. These include both conventional CNN architectures and hybrid models that integrate clustering, feature engineering, and ensemble techniques. For example, Kaur and Devendran [35] developed a semi-automatic system for detecting grape leaf diseases by combining K-Means clustering optimized using Grey Wolf Optimization (GWO), and hybrid feature extraction consisting of Law's Mask, GLCM, LBP, and Gabor features. The model was classified using an ensemble classifier and recorded an accuracy of 95.69%. A similar approach was taken by Javidan et al [36], who also used

the K-Means clustering method combined with a machine learning algorithm, resulting in an accuracy of 98.97%. Meanwhile, Ishengoma and Lyimo [37] used a Convolutional Neural Network (CNN) for feature extraction and a Random Forest for classification. In their experiments, features extracted from the VGG-16 and InceptionV3 architectures gave the highest accuracy of 95.34%. The combined fine-tuning-based model has also been tested in the literature. Ji et al [2] introduced a united model approach by combining ResNet50 and InceptionV3 as a fine-tuning technique, which achieved an accuracy of 98.57%. Furthermore, Canghai et al [6] introduced GC-MobileNet, which combines the MobileNet architecture with the Ghost convolution algorithm, and achieved an accuracy of 98.63%.

In parallel, modern deep CNN architectures have also been benchmarked extensively. EfficientNetB0, for example, has demonstrated superior accuracy in several studies, reaching up to 99.02% [21], consistently outperforming InceptionV3 in various tasks [8], [38]. EfficientNetB5 and B7 have also shown strong results (96.07% and 96.39%), though B0 is often preferred for its efficiency [19]. VGG16, though widely applied, often trails behind, achieving 96.29% [39], while ResNet50 and its deeper variants like ResNet152V2 offer improved generalization and reach up to 97.06% [20], [39], [40]. DenseNet121 has demonstrated exceptional accuracy (99.12%), and MobileNetV2 has also performed well, achieving 99% while remaining suitable for lightweight applications [41], [42].

Compared with those studies, the model in this study shows the highest accuracy of 99.09% on a 75-epoch configuration, accompanied by high and balanced precision, recall, and F1-score values. The advantages of this model lie in the efficiency of the architecture and the stability of performance as the epoch increases, without requiring the integration of optimization algorithms or complex hybrid models. Thus, this study's approach produces more accurate classification performance than previous models and offers a more straightforward, efficient, and reproducible solution for image-based plant disease detection systems.

From an application standpoint, the proposed model can be adapted for real-time deployment in mobile-based diagnostic systems or integrated into drone-based platforms for vineyard monitoring. The relatively lightweight nature of InceptionV3 and its compatibility with transfer learning pipelines make it well-suited for scalable, real-time plant health monitoring systems in precision agriculture.

5. CONCLUSION

This study has successfully developed and evaluated a deep learning-based classification model using three different training scenarios based on the number of epochs, namely 25, 50, and 75 epochs. With a batch size configuration of 32 and a learning rate of 0.001, the model shows a consistent increase in performance as the number of epochs increases. From the evaluation results using a confusion matrix, an accuracy of 98.64% was obtained at 25 epochs, 98.78% at 50 epochs, and 99.09% at 75 epochs. Adding epochs has improved the model's generalization ability to test data. The most common misclassification errors occurred between the Black Rot and ESCA classes, but these errors decreased significantly at higher epochs. The higher the number of epochs, the better the precision, recall, accuracy, and F1-score results. The configuration with 75 epochs provided the best performance, with an accuracy value reaching 99.09% and a perfect balance between metrics, without showing symptoms of overfitting. These findings indicate that training the model for a sufficient number of cycles contributes significantly to the stability and accuracy of the classification. Therefore, 75 epochs are recommended as the optimal parameter in model training for this study.

The implications of this research are significant for the field of Computer Science, particularly in the subdomains of computer vision and precision agriculture. The implementation of transfer learning using InceptionV3 offers a scalable and efficient approach to automated plant disease detection, which can contribute to increasing agricultural productivity and supporting smart farming initiatives.

Future research should consider expanding the dataset using field acquired images to improve robustness under real world conditions. Additionally, integrating this model into mobile or edge computing devices and evaluating its performance in real-time scenarios could further validate its practicality for on-site deployment.

CONFLICT OF INTEREST

The authors declares that there is no conflict of interest between the authors or with research object in this paper.

ACKNOWLEDGEMENT

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

REFERENCES

- [1] M. Fraiwan, E. Faouri, and N. Khasawneh, "Multiclass Classification of Grape Diseases Using Deep Artificial Intelligence," *Agriculture*, vol. 12, no. 10, p. 1542, Sep. 2022, doi: 10.3390/agriculture12101542.
- [2] M. Ji, L. Zhang, and Q. Wu, "Automatic Grape Leaf Diseases Identification Via UnitedModel Based on Multiple Convolutional Neural Networks," *Inf. Process. Agric.*, vol. 7, no. 3, pp. 418–426, Sep. 2020, doi: 10.1016/j.inpa.2019.10.003.
- [3] E. Osama, E. A., D. A. Mohammad, H. Mosab, and G. Jianmin, "Rapid Grapevine Health Diagnosis Based on Digital Imaging and Deep Learning," *Plants*, 2024, doi: 10.3390/plants13010135.
- [4] G. Salvador, H. Inés, C. S., B. I., D. N. Ana, and T. J., "Deep Learning for the Differentiation of Downy Mildew and Spider Mite in Grapevine Under Field Conditions," *Comput. Electron. Agric.*, 2021, doi: 10.1016/j.compag.2021.105991.
- [5] A. Alkan, M. Ü. Abdullah, H. O. Abdullah, M. Assaf, and H. Zhou, "A Smart Agricultural Application: Automated Detection of Diseases in Vine Leaves Using Hybrid Deep Learning," *Turk. J. Agric. For.*, vol. 45, no. 6, pp. 717–729, Dec. 2021, doi: 10.3906/tar-2007-105.
- [6] W. Canghai, G. Xingxiang, X. Huanliang, and H. Huixin, "Fine-Grained Recognition of Grape Leaf Diseases Based on Transfer Learning and Convolutional Block Attention Module," *Appl. Soft Comput.*, vol. 172, p. 112896, Mar. 2025, doi: 10.1016/j.asoc.2025.112896.
- [7] R. Mandal and M. K. Gouda, "Vineyard Vigilance: Harnessing Deep Learning for Grapevine Disease Detection," vol. 7, no. 1, Agustus 2024, doi: <https://doi.org/10.59720/23-251>.
- [8] P. Kaushik and P. Sharma, "Deep Learning-Based Classification of Grape Leaf Diseases using EfficientNetB0 and InceptionV3 for Enhanced Vineyard Management," in *2024 2nd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, Nov. 2024, pp. 892–897. doi: 10.1109/ICAICCIT64383.2024.10912256.
- [9] M. A. Ebrahimi, M. H. Khoshtaghaza, S. Minaei, and B. Jamshidi, "Vision-Based Pest Detection Based on Svm Classification Method," *Comput. Electron. Agric.*, vol. 137, pp. 52–58, May 2017, doi: 10.1016/j.compag.2017.03.016.
- [10] D. W. Girmaw and T. W. Muluneh, "Deep Convolutional Neural Network Model for Classifying Common Bean Leaf Diseases," *Discov. Artif. Intell.*, vol. 4, no. 1, p. 96, Nov. 2024, doi: 10.1007/s44163-024-00212-6.
- [11] K. P. Panigrahi, H. Das, A. K. Sahoo, and S. C. Moharana, "Maize Leaf Disease Detection and Classification Using Machine Learning Algorithms," in *Progress in Computing, Analytics and Networking*, H. Das, P. K. Pattnaik, S. S. Rautaray, and K.-C. Li, Eds., Singapore: Springer, 2020, pp. 659–669. doi: 10.1007/978-981-15-2414-1_66.
- [12] X. Zhou, W. S. Lee, Y. Ampatzidis, Y. Chen, N. Peres, and C. Fraisse, "Strawberry Maturity Classification from UAV and Near-Ground Imaging Using Deep Learning," *Smart Agric. Technol.*, vol. 1, p. 100001, Dec. 2021, doi: 10.1016/j.atech.2021.100001.

- [13] M. Sholihin, M. F. M. Fudzee, and M. N. Ismail, "AlexNet-Based Feature Extraction for Cassava Classification: A Machine Learning Approach," *Baghdad Sci. J.*, vol. 20, no. 6(Suppl.), Art. no. 6(Suppl.), Dec. 2023, doi: <https://10.21123/bsj.2023.9120>.
- [14] M. Sholihin, M. F. M. Fudzee, M. N. Ismail, E. N. Wati, M. S. Arshad, and T. Gusman, "Modified Alexnet Architecture for Classification of Cassava Based on Leaf Images," *JOIV Int. J. Inform. Vis.*, vol. 8, no. 3, Aug. 2024, doi: <https://10.62527/joiv.8.3.2966>.
- [15] B. Liu, Z. Ding, L. Tian, D. He, S. Li, and H. Wang, "Grape Leaf Disease Identification Using Improved Deep Convolutional Neural Networks," *Front. Plant Sci.*, vol. 11, Jul. 2020, doi: [10.3389/fpls.2020.01082](https://doi.org/10.3389/fpls.2020.01082).
- [16] S. Zhang, S. Zhang, C. Zhang, X. Wang, and Y. Shi, "Cucumber Leaf Disease Identification with Global Pooling Dilated Convolutional Neural Network," *Comput. Electron. Agric.*, vol. 162, pp. 422–430, Jul. 2019, doi: [10.1016/j.compag.2019.03.012](https://doi.org/10.1016/j.compag.2019.03.012).
- [17] K. Z. Thet, K. K. Htwe, and M. M. Thein, "Grape Leaf Diseases Classification using Convolutional Neural Network," in *2020 International Conference on Advanced Information Technologies (ICAIT)*, Nov. 2020, pp. 147–152. doi: [10.1109/ICAIT51105.2020.9261801](https://doi.org/10.1109/ICAIT51105.2020.9261801).
- [18] Z. Tang, J. Yang, Z. Li, and F. Qi, "Grape Disease Image Classification Based on Lightweight Convolution Neural Networks and Channelwise Attention," *Comput. Electron. Agric.*, vol. 178, p. 105735, Nov. 2020, doi: [10.1016/j.compag.2020.105735](https://doi.org/10.1016/j.compag.2020.105735).
- [19] S. G. Subramanya and A. Parkavi, "Performance Evaluation of EfficientNet Models in Grape Leaf Disease Classification," presented at the International Conference on Intelligent Systems and Computational Networks, ICISCN 2025, 2025. doi: [10.1109/ICISCN64258.2025.10934407](https://doi.org/10.1109/ICISCN64258.2025.10934407).
- [20] H.-T. Vo, K. C. Mui, N. N. Thien, P. P. Tien, and H. L. Le, "Optimizing Grape Leaf Disease Identification Through Transfer Learning and Hyperparameter Tuning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 2, pp. 922–931, 2024, doi: [10.14569/IJACSA.2024.0150293](https://doi.org/10.14569/IJACSA.2024.0150293).
- [21] A. K. Uttam, "Grape Leaf Disease Prediction Using Deep Learning," presented at the Proceedings - International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2022, 2022, pp. 369–373. doi: [10.1109/ICAAIC53929.2022.9792739](https://doi.org/10.1109/ICAAIC53929.2022.9792739).
- [22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2818–2826. doi: [10.1109/CVPR.2016.308](https://doi.org/10.1109/CVPR.2016.308).
- [23] T. Hidayat, I. A. Astuti, A. Yaqin, A. P. Tjilen, and T. Arifianto, "Grouping of Image Patterns Using Inceptionv3 For Face Shape Classification," *JOIV Int. J. Inform. Vis.*, vol. 7, no. 4, pp. 2336–2343, Dec. 2023, doi: [10.62527/joiv.7.4.1743](https://doi.org/10.62527/joiv.7.4.1743).
- [24] K. Kamaludin, W. I. Rahayu, and M. Y. H. Setywan, "Transfer Learning to Predict Genre Based on Anime Posters," *J. Tek. Inform. Jutif*, vol. 4, no. 5, Art. no. 5, Oct. 2023, doi: [10.52436/1.jutif.2023.4.5.860](https://doi.org/10.52436/1.jutif.2023.4.5.860).
- [25] D. R. A. N. Pratama and Y. Azhar, "Convolutional Neural Network for COVID-19 Detection Using InceptionV3 Transfer Learning," *J. Tek. Inform. Jutif*, vol. 6, no. 2, Art. no. 2, Apr. 2025, doi: [10.52436/1.jutif.2025.6.2.4094](https://doi.org/10.52436/1.jutif.2025.6.2.4094).
- [26] V. Ahuja, S. Vats, and R. Sharma, "Pre-trained InceptionV3 for Efficient Tuberculosis Detection," *2024 15th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT*, pp. 1–5, Jun. 2024, doi: [10.1109/ICCCNT61001.2024.10725290](https://doi.org/10.1109/ICCCNT61001.2024.10725290).
- [27] A. A. Mahmood, S. Sadeq, Y. I. Aljanabi, and A. H. Sabry, "Developing a Convolutional Neural Network for Classifying Tumor Images Using Inception V3," *East.-Eur. J. Enterp. Technol.*, vol. 3, no. 9 (123), Art. no. 9 (123), Jun. 2023, doi: [10.15587/1729-4061.2023.281227](https://doi.org/10.15587/1729-4061.2023.281227).
- [28] R. Singh, N. Sharma, P. Aggarwal, M. Singh, and K. R. Chythanya, "InceptionV3 in Medical Imaging: Enhancing Precision in Acute Lymphoblastic Leukaemia Diagnosis," in *2024 2nd International Conference on Computer, Communication and Control (IC4)*, Feb. 2024, pp. 1–6. doi: [10.1109/IC457434.2024.10486481](https://doi.org/10.1109/IC457434.2024.10486481).
- [29] W. Wahyuningsih, G. S. Nugraha, and R. Dwiyanaputra, "Classification of Dental Caries Disease in Tooth Images Using a Comparison of Efficientnet-B0, Mobilenetv2, Resnet-50, Inceptionv3 Architectures," *J. Tek. Inform. Jutif*, vol. 5, no. 4, Art. no. 4, Jul. 2024, doi: [10.52436/1.jutif.2024.5.4.2187](https://doi.org/10.52436/1.jutif.2024.5.4.2187).

-
- [30] S. Das, H. B. Karna, S. Das, and R. Hazra, "Disease Detection in Plant Leaves Using Transfer Learning," presented at the Proceedings - 1st International Conference on Electronics, Communication and Signal Processing, ICECSP 2024, 2024. doi: 10.1109/ICECSP61809.2024.10698296.
- [31] R. Tamilmani and R. R. Khan, "Investigation of transfer learning for plant disease detection across different plant species," in *Applications of Mathematics in Science and Technology: International Conference on Mathematical Applications in Science and Technology*, 2025, pp. 440–444. doi: 10.1201/9781003606659-83.
- [32] P. Dubey, V. Kumari, A. K. Sharma, and G. Sheoran, "Quantitative Analysis of Transfer Learning in Plant Disease Classification," presented at the Lecture Notes in Networks and Systems, 2022, pp. 909–918. doi: 10.1007/978-981-19-2130-8_71.
- [33] B. Imane and A. Mohamed, "Grape and Apple Plant Diseases Detection Using Enhance DenseNet121 Based Convolutional Neural Network," presented at the Lecture Notes in Networks and Systems, 2024, pp. 213–226. doi: 10.1007/978-3-031-47672-3_22.
- [34] R. Mandal and M. Kheir Gouda, "Vineyard Vigilance: Harnessing Deep Learning for Grapevine Disease Detection," *J. Emerg. Investig.*, vol. 7, pp. 1–9, 2024, doi: 10.59720/23-251.
- [35] N. Kaur and V. Devendran, "A Novel Framework for Semi-Automated System for Grape Leaf Disease Detection," *Multimed. Tools Appl.*, vol. 83, no. 17, pp. 50733–50755, May 2024, doi: 10.1007/s11042-023-17629-3.
- [36] S. M. Javidan, A. Banakar, K. A. Vakilian, and Y. Ampatzidis, "Diagnosis of Grape Leaf Diseases Using Automatic k-Means Clustering and Machine Learning," *Smart Agric. Technol.*, vol. 3, p. 100081, Feb. 2023, doi: 10.1016/j.atech.2022.100081.
- [37] F. S. Ishengoma and N. N. Lyimo, "Ensemble Model for Grape Leaf Disease Detection Using Cnn Feature Extractors and Random Forest Classifier," *Heliyon*, vol. 10, no. 12, p. e33377, Jun. 2024, doi: 10.1016/j.heliyon.2024.e33377.
- [38] A. P. S. Yadav, N. Thapliyal, M. Aeri, V. Kukreja, and R. Sharma, "Advanced Deep Learning Approaches: Utilizing VGG16, VGG19, and ResNet Architectures for Enhanced Grapevine Disease Detection," presented at the 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2024, 2024. doi: 10.1109/ICRITO61523.2024.10522276.
- [39] Y. Li, Z. Liu, Y. Jiang, and T. T. Toe, "Transfer Learning in Grape Disease Leaf Detection Based on Convolutional Neural Network," presented at the 2023 4th International Seminar on Artificial Intelligence, Networking and Information Technology, AINIT 2023, 2023, pp. 51–57. doi: 10.1109/AINIT59027.2023.10212855.
- [40] D. Kala, D. Punia, G. Sikka, and K. Sikka, "A Comprehensive Analysis of Various Deep Learning Based Multi Class Plant Disease Classification Techniques," presented at the 1st International Conference on Pioneering Developments in Computer Science and Digital Technologies, IC2SDT 2024 - Proceedings, 2024, pp. 107–112. doi: 10.1109/IC2SDT62152.2024.10696395.
- [41] A. Goyal and K. Lakhwani, "Integrating advanced deep learning techniques for enhanced detection and classification of citrus leaf and fruit diseases," *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-97159-0.
- [42] F. A. Maulana, K. Kertarajasa, Y. S. Yasa, S. A. Sari, and M. D. Sulistiyo, "Grapevine Leaves Classification Using Various CNN Model," presented at the Proceedings - 11th International Conference on Information Technology, Computer and Electrical Engineering, ICITACEE 2024, 2024, pp. 224–229. doi: 10.1109/ICITACEE62763.2024.10761961.
-