Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 24, No. 2, March 2025, pp. 333∼346
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v24i2.4137

Integration of Image Enhancement Technique with DenseNet201 Architecture for Identifying Grapevine Leaf Disease

Rudi Kurniawan , Lukman Sunardi
Universitas Bina Insan, Lubuklinggau, Indonesia

Article Info

Article history:

Received June 13, 2024
Revised February 22, 2025
Accepted March 12, 2025

Keywords:

Convolutional Neural Network.,
DenseNet201;
Deep learning;
Image Classification;
Image Enhancement;
Leaf Disease Detection.

ABSTRACT

Early detection of grapevine leaf diseases is crucial for maintaining both the quality and quantity of grape production. Manual identification methods are often ineffective and prone to errors. This research aims to develop a precise and efficient method for classifying grapevine leaf diseases using Contrast Limited Adaptive Histogram Equalization (CLAHE) and the DenseNet201 Deep Convolutional Neural Network (DCNN) architecture. The research methodology involves collecting a dataset of grapevine leaf images affected by black measles, black rot, and leaf blight alongside healthy leaves. Following this, preprocessing is conducted using the CLAHE technique to enhance image quality. Then, the processed data is trained with DenseNet201. Evaluation results indicate that the proposed model achieves an overall accuracy of 99.61%, with high precision, recall, and F1-score values across all disease classes. Receiver Operating Characteristic (ROC) curve analysis shows an Area Under the Curve (AUC) of 1.00 for each class, reflecting excellent discriminatory ability. The loss and accuracy curves illustrate consistent model performance without signs of overfitting. Additionally, the confusion matrix confirms very low classification error rates. The developed model is effective and reliable for identifying grapevine leaf diseases. Future research will focus on enhancing the dataset by incorporating more data optimizing hyperparameters, and developing field applications for real-time use.

Corresponding Author:

Rudi Kurniawan, +62 853 6811 7030,
Faculty of Engineering Science Department of Computer System Engineering,
Universitas Bina Insan, Lubuklinggau, Indonesia,
Email: rudi.kurniawan@univbinainsan.ac.id.

How to Cite:
R. Kurniawan and L. Sunardi, ”Integration of Image Enhancement Technique with DenseNet201 Architecture for Identifying Grapevine Leaf Disease”, *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 24, No.2, pp. xxx-xxx, March, 2025.

1. INTRODUCTION

Leaf disease classification is one of the major areas in modern agriculture aimed at identifying and effectively managing plant diseases. In this context, grapes are a high-value crop that is susceptible to various types of leaf diseases. Early detection and accurate classification of grapevine leaf diseases can benefit the growers by taking appropriate preventive measures, reducing yield losses, and improving production quality [1]. Grapevine leaf diseases often have similar visual symptoms, such as leaf discoloration, spotting, or deformation [2]. This makes manual classification by plant experts a challenging and time-consuming task. In addition, varying lighting conditions, shadows, and image noise can affect classification accuracy [3]. Therefore, an automated method is needed that can handle this variability efficiently and accurately when using deep learning techniques focused on image classification [4–8].

In developing this grapevine leaf disease classification, several relevant previous studies are discussed to provide an overview of image processing and deep learning classification techniques used in plant disease detection. In the research by [9], the Adaptive Snake Model technique was used to segment and identify grape leaf diseases. This model includes two segmentation phases: joint segmentation for fast segmentation and absolute segmentation for better accuracy. Evaluations were performed using the PlantLevel and PlantVillage datasets, and the results showed that these models outperformed various other methods in terms of performance metrics such as Manhattan, Peak Signal-to-Noise Ratio (PSNR), Recall, Dice, and Jaccard. The research developed by [10] combines the LoRa protocol with a deep learning-based computer vision system to transmit and identify grape leaf diseases using low-resolution images. The pre-processing uses Low Power Longer Range (LoRa) parameters, fine-tuning of the Convolutional Neural Network (CNN) model, and the Grad-CAM technique to visualize the CNN results. The evaluation shows that this model can transmit images within LoRa bandwidth limitations and efficiently identify grape leaf diseases. The research conducted by [11], reviewed various computer vision and soft computing techniques for automatically detecting plant diseases using leaf images. They evaluated modern feature extraction techniques on various plant categories and concluded that computer vision techniques can detect and classify plant diseases effectively.

Research conducted by [12], recommended an automatic classification method for grape leaf diseases using image analysis and the Back Propagation Neural Network (BPNN) approach. The pre-processing technique includes Wiener filtering, wavelet transform for denoising, segmentation using the Otsu method, and a morphological algorithm to improve the lesion's shape. Features such as perimeter, area, circularity, rectangularity, and shape complexity are extracted. The results show that the BPNN model has high accuracy in detecting five grape leaf diseases. Research conducted by [13], developed a Generative Adversarial Network (GAN) technique for image data augmentation of grape leaf diseases. The Leaf GAN model combines a generator with degressive channels and an efficient discriminator. The results show that the Leaf GAN model produces images with prominent lesions and improves the accuracy of disease identification by CNN up to 98.70% over the Xception model. In research conducted by [14], an integrated CNN architecture called the United model, was developed to detect grape leaf diseases such as black rot, esca, and isariopsis leaf spot. This model combines multiple CNNs to extract complementary features, resulting in an average validation accuracy of 99.17% and a testing accuracy of 98.57%.

In research conducted by [15], this study compared DCNN models such as AlexNet, GoogLeNet, and ResNet-18 integrated with Recurrent-Convolutional Neural Network (R-CNN) to identify three grape leaf diseases. In the experiments, AlexNet achieves the highest accuracy of 95.65%, followed by GoogleNet with an accuracy of 92.29%, and ResNet-18 with an accuracy of 89.49%. In the work of [16], fine-grained GAN is used to augment local spot area data in grape leaf images. This method is combined with faster R-CNN to detect local spot areas, resulting in an identification accuracy of 96.27% on the ResNet-50 model. In the research conducted by [17] To improve the accuracy of grape leaf disease detection, the Squeeze-and-Excitation Networks (SE), Efficient Channel Attention (ECA), and Convolutional Block Attention Module (CBAM) attention mechanisms in the Faster R-CNN, You Only Look Once (YOLOx), and Single Shot Multibox Detector (SSD) models are used. The results show that YOLOx+ECA has the highest accuracy and the best real-time performance.

Research conducted by [18], proposed the GrapeGAN architecture to produce high-resolution images of grape leaf diseases. GrapeGAN combines a U-net-based generator with residual blocks and a reorganization method, resulting in an identification accuracy of up to 96.13% on VGG16 and InceptionV1 models. Meanwhile, research by [19], developed a lightweight CNN model, GrapeNet, which uses residual blocks, Residual Feature Fusion Blocks (RFFBs), and CBAM to identify different stages of grape leaf disease symptoms. GrapeNet achieves the best accuracy compared to other models with only 2.15 million parameters, demonstrating high efficiency and performance. Research conducted by [20], SSD, Faster R-CNN, and YOLOx were used with attention mechanisms to increase the accuracy of grape leaf disease detection. YOLOx+ECA shows the highest accuracy and best real-time performance, providing valuable insights for automated disease detection in agricultural production.

Although several studies have used deep learning techniques and image processing to identify grape leaf diseases, several gaps must be filled. Many studies still face challenges regarding input image quality, especially in poor lighting conditions or low-resolution images. Data augmentation techniques such as GANs can help, but there is still room to improve the sharpness and clarity of the resulting images. In addition, many models require large and diverse training data to achieve high accuracy, which is not always available. The application of attention mechanisms in detection models shows promising results, but more research is needed to optimize the combination of these techniques with different CNN models. Finally, most studies focus on detecting single or limited diseases, while accurately detecting multiple diseases in a single model remains challenging.

To overcome these challenges, we propose using image enhancement techniques with the Contrast Limited Adaptive Histogram Equalization (CLAHE) method and classification using Deep Convolutional Neural Network (DCNN), specifically the DenseNet-201 model. CLAHE adaptively enhances image contrast by dividing the image into small tiles and applying histogram equalization to each tile, thus reducing the effects of noise and shadows while enhancing details in leaf disease images. DCNNs are a class of artificial neural networks that are highly effective in image recognition tasks. We choose DenseNet-201, a deep network architecture known for its parameter efficiency and ability to capture intricate image features. The dense connectivity pattern of DenseNet-201 has demonstrated superior performance in image classification tasks, particularly for complex features often encountered in plant disease detection [21]. This architecture facilitates feature reuse and propagation throughout the network, effectively mitigating the vanishing gradient problem, which allows for training deeper networks with fewer parameters compared to architectures like VGGNet or ResNet [22]. This is achieved through dense connections where each layer is connected to every subsequent layer, further enhancing information flow and addressing the vanishing gradient issue.

The expected contributions of this research include: (i) Applying the CLAHE technique to grapevine leaf images to increase visual contrast and detail, thereby helping to reduce the effects of shadows and noise, as well as clarifying disease symptoms on the leaves, (ii) Using DCNN with the DenseNet-201 architecture, which is known for its high parameter efficiency and ability to capture important features from images, and (iii) Integration of image enhancement techniques (CLAHE) with a powerful classification model (DenseNet-201) capable of identifying grapevine leaf diseases with high accuracy.

2. RESEARCH METHOD

This section provides a concise description of the proposed method, an explanation of the dataset used, preprocessing techniques (CLAHE), a detailed description of the DCNN model architecture (DenseNet201), and an overall evaluation of the model. Figure 1 illustrates the proposed method for grapevine leaf disease classification.

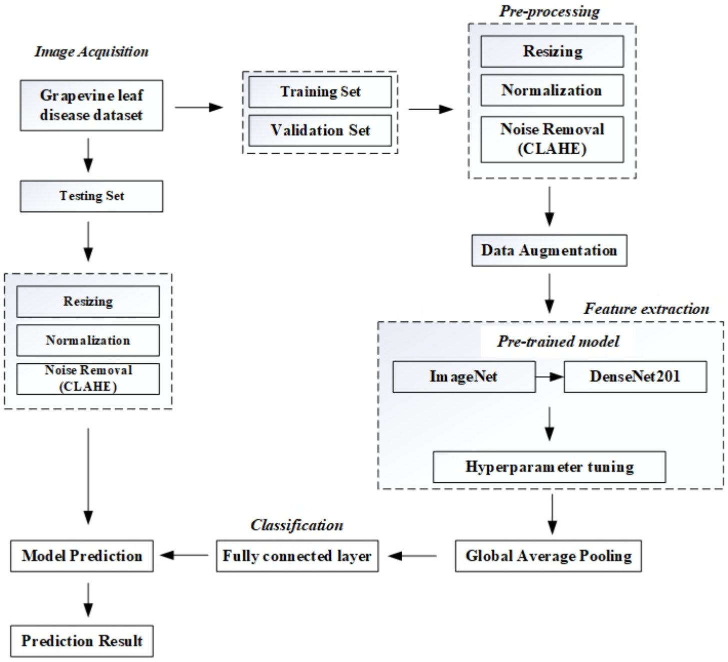

Figure 1. Proposed method

2.1. Data Acquisition

The data utilized in this study consists of grape leaf disease images sourced from the Plant Village dataset, comprising a total of 14.132 images categorized into four distinct classes: healthy, black rot, black measles, and leaf blight [23]. This diverse dataset is crucial for training and validating our proposed model, ensuring it can effectively differentiate between various conditions affecting grape leaves. Table 1 illustrates the distribution of images across these classes, providing insight into the representation of each category within the dataset, which is essential for achieving balanced and reliable classification results.

Table 1. Image Data Distribution

Classes		**Amount of Images**
Diseases	black measles	1383
	black rot	11390
	leaf blight	889
Healthy leaves		470
Total		**14132**

2.2. Pre-processing

Preprocessing is an important step in processing image data before it is used in deep-learning models [24, 25]. This process involves several techniques to improve the quality of the image, reduce noise, and supply data for model training. The three main techniques used in pre-processing are resizing, normalization, and image enhancement with CLAHE.

All grape leaf images were resized to 224×224 pixels, which matches the input size required by Deep Convolutional Neural Network (DCNN) architectures such as DenseNet201. The resized image is then normalized by dividing the pixel value by 255. This normalization changes the pixel value range from 0-255 to 0-1, which helps to speed up the model training process and increase convergence stability.

CLAHE is applied to the image to increase local contrast [26]. This method divides the image into several small parts (tiles), performs adaptive histogram equalization on each part, and then recombines them. This technique helps to clarify details in the image, especially in areas of low contrast, which is important for identifying disease characteristics on grape leaves. Figure 2 shows the results of image enhancement using CLAHE.

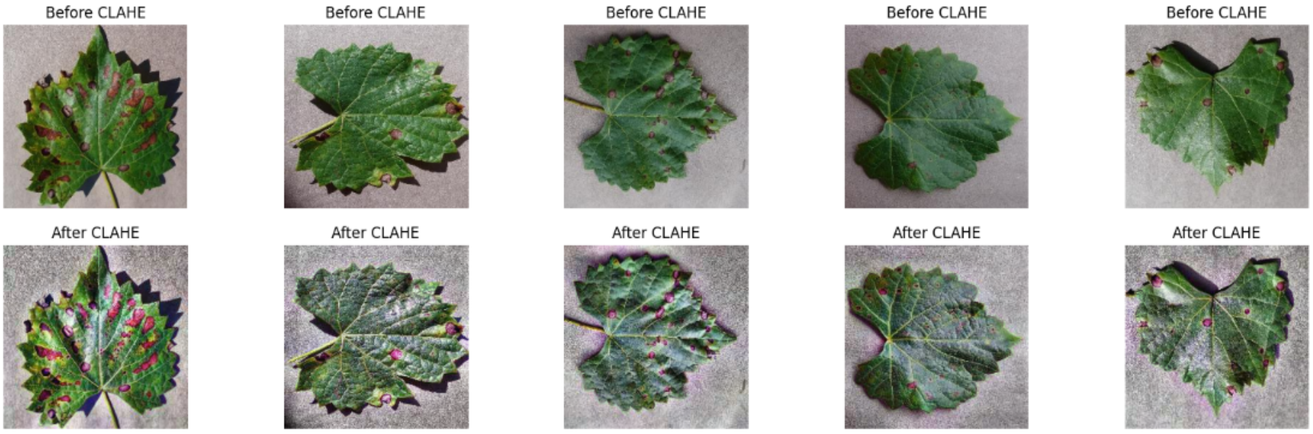

Figure 2. Image enhancement using CLAHE

2.3. Image Augmentation

Image augmentation is a technique that enriches image datasets by applying various transformations to the original images. This approach significantly boosts both the quantity and diversity of the dataset [27]. The primary objective is to enhance the model's robustness against variations that may arise in real-world data, thereby improving its ability to generalize effectively. By expanding the diversity of the training data, the model becomes better equipped to recognize familiar patterns rather than simply memorizing the training examples, which helps mitigate the risk of overfitting. The data augmentation techniques utilized in this study are detailed in Table 2.

Table 2. Image Augmentation

Technique	Value	Description
Rotation range	10	The image is randomly rotated within a range of 10 degrees (either CW or CCW).
Width shift range	0.02	The image is shifted horizontally (left or right) by a maximum of 2% of the image width.
Height shift range	0.02	The image is shifted vertically (up or down) by a maximum of 2% of the image height.
Zoom	0.05	The image is randomly enlarged or reduced within a 5% range.
Horizontal	TRUE	The image is mirrored horizontally.

The transformation of each technique in data image augmentation is clearly shown in Figure 3.

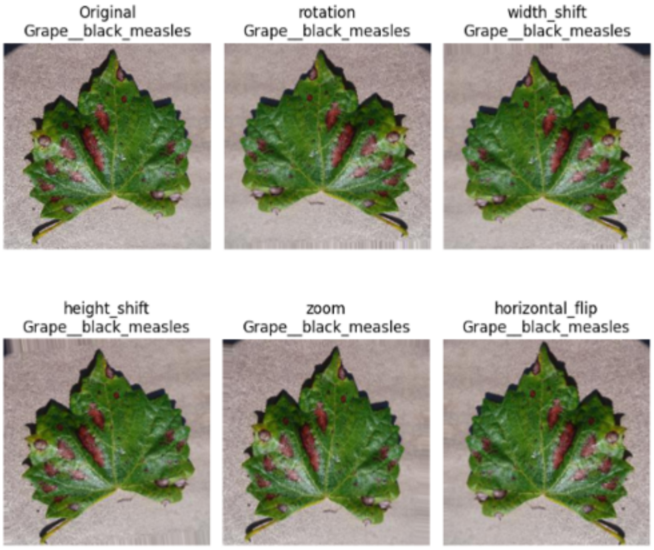

Figure 3. Image transformation with augmentation

2.4. Pre-trained with DenseNet201 model

DenseNet201 is a variant of the DenseNet (Densely Connected Convolutional Networks) architecture designed to improve information flow and gradients throughout the network by using direct connections between each layer and every other layer [28]. In DenseNet, every layer receives input from all preceding layers, enabling feature reuse and mitigating the issue of the vanishing gradient. DenseNet201 consists of a series of blocks, called Dense Blocks, connected by Transition Layers. Each Dense Block contains several densely connected convolutional layers. Table 3 below shows details of the model architecture based on the output of DenseNet201 and additional layers for classification.

Table 3. Summary of The Model

Layer (type)	Output shape	Param#
Input layer	$(224\times224\times3)$	0
Densenet201 (functional)	$(7\times27\times1920)$	18321984
Batch normalization	$(7\times27\times1920)$	7680
Global Average Pooling	1920	0
Dense	128	245888
Output layer	4	516
Total params		18576068
Trainable params		250244
Non-trainable params		18325824

Input layer with size (224, 224, 3), DenseNet201 as a pre-trained model with size (7, 7, 1920) and the number of parameters of 18,321,984 Dense connections between convolution layers for feature reuse. Batch normalization for DenseNet201 output helps stabilize and speed up training with 7.680 parameters. Global Average Pooling2D is used to flatten the convolution output into a one-dimensional vector with output shape (1920) without additional parameters.

The results of this 1D vector are passed to the 128-unit fully connected layer using ReLU activation for nonlinearity with a total of 245.888 parameters. The final dense layer for classification into four classes with softmax activation for probability output with 516 parameters. The total number of parameters produced was 18,576,068, of which 250.244 were trainable parameters (only new layers were added), and 18,325,824 were non-trainable parameters (parameters from DenseNet201 that were pre-trained and not re-trained).

2.5. Evaluation

2.5.1. Confusion Matrix

The Confusion Matrix (CM) is the most commonly used method to figure out the achievement of classification models in machine learning [29]. CM represents the comparison between predicted and actual values from the classification model The CM can be used to calculate several important metrics such as f1-score, accuracy, recall, and precision, which help to figure out the performance of the grapevine leaf disease model. Table 4 presents the configuration of the CM.

Table 4. Confusion Matrix

	Actually positive	**Actually negative**
Predicted Positive	True Positives (TP)	False Positives (FN)
Predicted Negative	False Negatives (FN)	True Negatives (TN)

Using these components, we can calculate several important metrics for evaluating Classification, including The accuracy of the model's predictions, which is calculated according to equation 1 and is represented as a percentage of correctly predicted samples. Precision is calculated as Equation 2 and displays the percentage of positive data samples accurately predicted by the model. Recall, also known as sensitivity or true positive rate, implies the percentage of positive samples that the model accurately predicts. Equation 3 can be used to calculate it. The F-1 Score describes the weighted average precision compared to the weighted average recall, which can be represented by Equation 4.

$$Acc = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{1}$$

$$pre = \frac{t_p}{t_p + f_p} \tag{2}$$

$$recall = \frac{t_p}{t_p + f_n} \tag{3}$$

$$f1 - score = 2\frac{pre * recall}{pre + recall} \tag{4}$$

2.5.2. Loss and Accuracy Graphs

Loss and accuracy graphs are crucial visual instruments for evaluating machine learning models, particularly in supervised learning tasks. The loss graph represents the model's performance in minimizing the discrepancy between predictions and actual values on the training data, where a steady decline indicates an effective learning process [30]. Meanwhile, the accuracy graph illustrates the model's ability to make correct predictions, with consistent improvement signifying enhanced performance. Analyzing these two graphs enables the identification of potential issues such as overfitting or underfitting.

Interpreting loss graphs involves observing a steady decline (indicating positive learning progress), a plateau (suggesting optimal performance or a local minimum), and an increase after several epochs (indicating overfitting). Comparing training and validation loss curves is important; close alignment between the curves indicates good generalization, while divergence suggests overfitting or underfitting. Similarly, interpreting accuracy graphs involves observing steady improvement, a plateau, and a decline. Comparing training and validation accuracy curves is also crucial for assessing the model's generalization ability.

Several strategies can be applied to address identified issues. Overfitting can be mitigated through regularization, dropout, data augmentation, or early stopping. Meanwhile, underfitting can be addressed by increasing model complexity, extending training duration, or performing feature engineering. By carefully analyzing loss and accuracy graphs, machine learning practitioners can optimize their model's performance and ensure good generalization to new data.

2.5.3. Receiver Operating Characteristic (ROC) Curve

The ROC curve is a graphical representation that evaluates the performance of a binary classification model by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various classification thresholds [31]. TPR measures the proportion of actual positives correctly identified by the model (sensitivity), while FPR measures the proportion of actual negatives incorrectly classified as positives. The ROC curve provides a comprehensive visualization of the model's ability to distinguish between positive and negative classes, unaffected by changes in class distribution or varying misclassification costs.

The Area under the Curve (AUC) of the ROC curve is a single summary metric that quantifies the model's overall performance. AUC ranges from 0 to 1, with higher values indicating better performance. An AUC of 0.5 suggests performance no better than random guessing, while an AUC of 1 indicates perfect classification. The ROC curve and AUC enable the comparison of performance across different classification models or model configurations and assist in selecting the optimal classification threshold based on the desired trade-off between sensitivity and specificity.

2.6. Experiment Set Up

In this experiment, the grapevine leaf disease dataset was split into three categories: training set, validation set, and testing set, with a ratio of 70%, 10%, and 20%. Figure 4 displays the distribution of each class in the training, validation, and testing datasets.

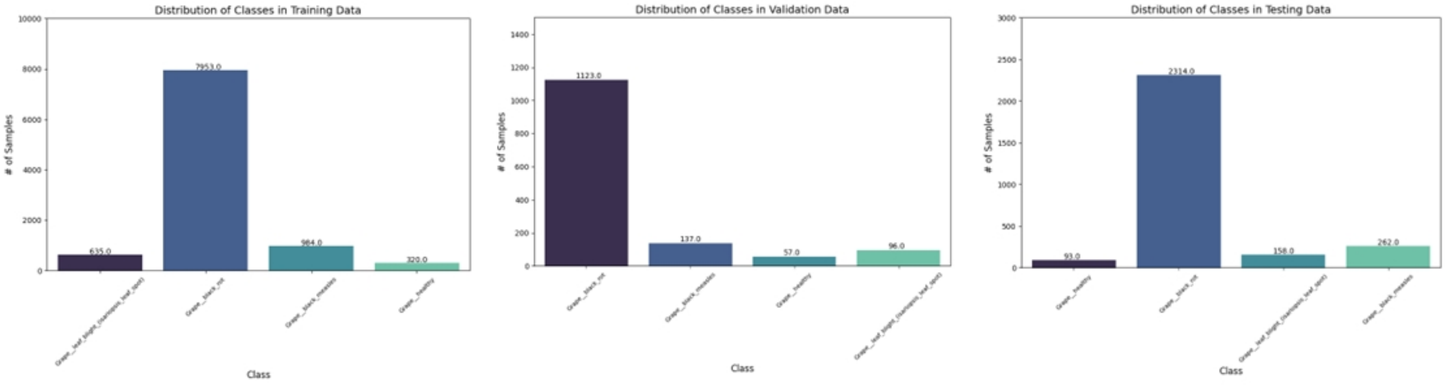

Figure 4. Data distribution for each class

The experiments in this research used the Python programming language and libraries, such as OpenCV, Sci-kit Learn, TensorFlow, and Keras. This research employed the following hardware and software configuration: a 9th generation Intel Core i7 processor, an NVIDIA GeForce GTX 1660 Ti graphics card, 16 GB of DDR4 RAM, the Windows 11 operating system, and the Python programming language version 3.10.2. Model training parameters included a batch size of 32, an initial learning rate of 0.001, a maximum of 50 epochs with early stopping implemented, and the Adam optimization algorithm to accelerate convergence and achieve optimal results. Table 5 lists the additional configuration parameters and hyperparameters for training.

Table 5. Software, Hardware, and Hyperparameter Environment

Configuration item	**Value**
CPU	core i7 gen 9th
GPU	NVIDIA GeForce GTX 1660 Ti
RAM	DDR4 16 GB
OS	Windows 11
Python	3.10.2
Batch size	32
Learning rate	0.001
Epoch	50
Early stopping	TRUE
Optimizer	Adam

3. RESULT AND ANALYSIS

3.1. Result

This section discusses the results of evaluating the grape leaf disease classification model using DenseNet201 with preprocessing using the CLAHE technique. The evaluation uses various metrics, including classification reports, confusion matrices, and Receiver Operating Characteristic (ROC) plots. The classification report provides detailed information about the model's performance in each class, including precision, recall, and F1 score. With this classification report, we can understand how well the model detects each type of disease on grape leaves and identify weaknesses in the predictions.

Figure 5 shows the changes in loss values as the model is trained. At the beginning of training (epoch 0 to about epoch 5), there was a very fast decrease in loss for both training and validation data. This shows that the model learns quickly from the data provided in the early stages of training. After around epoch 5, the loss value for the training data continues to decrease at a slower rate and finally stabilizes near zero. This shows that the model continues to improve its predictions and reaches convergence, where the decrease in loss becomes very small from one epoch to the next. The loss value for the validation data also shows a consistent decrease, but slower than the training data, stabilizing around epoch 10-15. This indicates that the model did not suffer from significant overfitting, i.e., it did not become so over-trained on the training data that it lost the ability to generalize to previously unseen data.

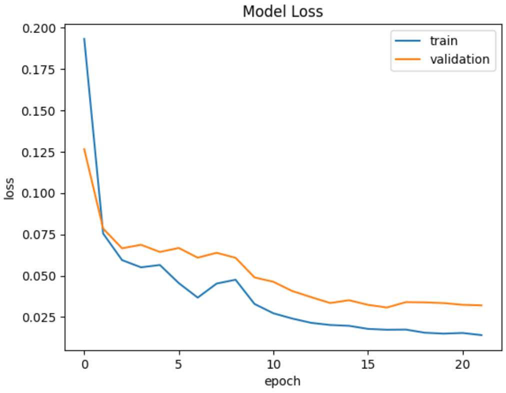

Figure 5. Loss value while training

The loss value for validation data is generally higher than the loss value for training data. This is a common pattern because the model is usually more accurate on data it has seen during training than on new (validation) data. The difference between training and validation loss is not very large, which indicates that the model exhibits strong generalization ability and does not suffer from overfitting the training data. At the end of the graph, both training and validation losses show stability, which means the model has reached a balance and has not experienced major changes in the last few epochs. This shows that the model has been trained sufficiently, and there is no need to continue training to avoid overfitting. Overall, the loss graph shows that the model has a good training performance, with a rapid decrease at the beginning of training and stability reached at the end. The loss value for the validation data is not too high compared to the training data, indicating that the model has good generalization capabilities and is not overfitting. This indicates that the training process was successful, and the resulting model performed well in the grape leaf disease classification task.

Overall, the loss graph shows that the model has a good training performance, with a rapid decrease at the beginning of training and stability reached at the end. The loss value for the validation data is not too high compared to the training data, indicating that the model has good generalization capabilities and is not overfitting. This indicates that the training process was successful, and the resulting model performed well in the grape leaf disease classification task.

Figure 6 shows the changes in accuracy values as the model is trained. At the beginning of training (epoch 0 to about epoch 5), there was a very rapid increase in accuracy for both training and validation data. This shows that the model learns quickly and improves its ability to correctly classify data early in training. After the initial increase, the accuracy for the training data continued to show small fluctuations but was generally stable and approached its maximum value (1.0) after about epoch 10. This indicates that the model had learned the patterning in the training data very well and consistently. The accuracy for the validation data also shows a stable pattern of improvement and is close to the maximum value, albeit with slight fluctuations. This demonstrates that the model possesses strong generalization capabilities and does not suffer from overfitting the training data.

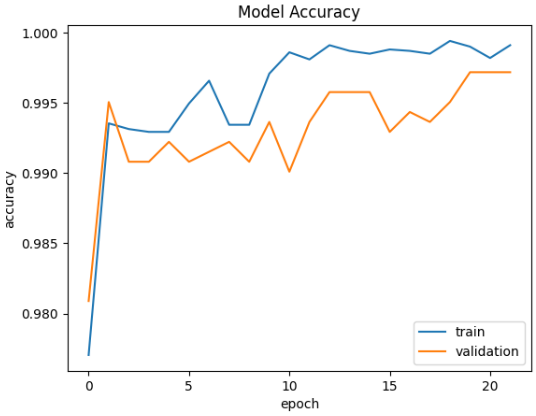

Figure 6. Accuracy value while training

The training and validation data accuracy is very close, especially after the first epoch. This shows that the model works well on data already seen (training data) and new data (validation data). The small differences that exist between training and validation accuracy are normal and indicate that the model does not experience significant overfitting. There were small fluctuations in the training and validation data accuracy values. These fluctuations may be due to variations in the data batches used during training, but overall the visible trend is one of improvement and stability. Overall, the accuracy graph shows that the model has excellent training performance, with accuracy increasing rapidly in the early stages and stabilizing near its maximum value in subsequent epochs. The high accuracy of validation data shows that the model has good generalization capabilities, so it can be relied upon to classify data that has never been seen before.

Table 6 displays the classification report outcomes for four categories of diseases observed on grape leaves. By examining the table, we can observe that the Precision value quantifies the accuracy of the model's positive predictions. In other words, the precision shows how many positive predictions are positive. The high precision values in all classes, especially "Black Rot" (0.9987) and "Leaf Blight" (1.0000), show that the model very rarely gives false positive results.

Table 6. Classification Report of Grapevine Leaf Disease Model

	Precision	**Recall**	**F1-Score**
Black measles	0.9924	0.9924	0.9924
Black rot	0.9987	0.9965	0.9976
Healthy	0.9388	0.9892	0.9634
Leaf blight	10.000	10.000	10.000
Accuracy			**0.9961**
Macro avg	0.9825	0.9945	0.9883
Weighted avg	0.9962	0.9961	0.9961

Recall is a measure that determines how effectively a model detects all positive instances. High recall values for all classes, especially for ”Leaf Blight” (1.0000) and ”Healthy” (0.9892), indicate that the model rarely misses true positive instances. The F1 score represents a balance between precision and recall. A good balance between precision and recall is indicated by high F1 scores in all classes, with ”Leaf Blight” (1.0000) achieving perfect performance. Support shows the actual number of instances of each class in the dataset. Most of the data consists of the class ”Black rot” (2314), indicating an imbalance in the data. However, the model still shows excellent performance in all classes. The Macro Average provides a simple metrics average across all classes, giving equal weight to each class without regard to frequency. The high Macro Average value (Precision: 0.9825, Recall: 0.9945, F1 Score: 0.9883) indicates that the model performance is consistent across classes. The weighted average takes into account the frequency of each class. Since the class ”black rot” dominates the data set, the weighted average is almost equal to the individual metrics of this class. The very high Weighted Average values (0.9961 for Precision, F1 Score, and Recall) confirm that the model behaves very well on the dataset as a whole.

Overall, the model performed excellently in classifying grape leaf diseases using CLAHE and DenseNet201. The almost perfect precision, recall, and F1 scores in all classes show that this model is very accurate and efficient in detecting leaf diseases. The comprehensive accuracy value of 99.61% confirms the reliability of this model in real practice. This shows that using the CLAHE preprocessing technique and DenseNet201 architecture is an effective choice for grape leaf disease classification.

Figure 7 shows a plot of the CM. The CM shows that the model has very high accuracy, especially for the classes ”Black rot” and ”Leaf blight,” where almost all instances are predicted correctly. The number of prediction errors is very low for all classes. For example, only two instances of ”Black measles” were incorrectly classified as ”Black rot” and vice versa. This error is very small compared to the number of correctly predicted instances. The ”healthy” class had a very low error rate, with only one instance misclassified as ”black rot.” This shows that the model can distinguish well between healthy and diseased leaves. There was no prediction error for the ”Blight” class, indicating that the model accurately recognizes this class.

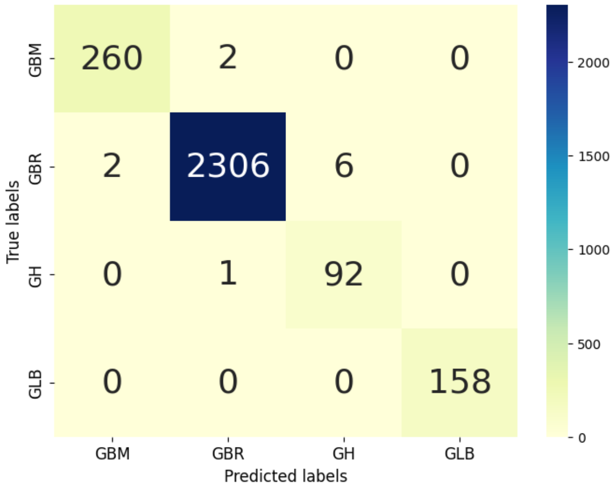

Figure 7. The CM of the model

Figure 8 displays the ROC curve of the model. The ROC (Receiver Operating Characteristic) graph shows the model’s performance in classifying four types of grape leaf conditions. All ROC curves for the four classes are very close to the upper left corner of the graph (true positive rate = 1, false positive rate = 0), indicating that the model has excellent classification performance for all classes. The AUC value for each class is 1.00, ranging from 0 to 1, where 1 indicates perfect classification performance. An AUC of 1.00 for each class indicates that the model is very good at distinguishing between positive and negative classes for all classes. In this graph, the FPR is very low (close to 0), while the TPR is very high (close to 1) for all classes. This means that the model has a very low error rate and a very high success rate in classifying the condition of grape leaves. The dashed diagonal line (from the lower left corner to the upper right corner) shows the model’s performance making random predictions (AUC = 0.5). All ROC curves are well above this line, indicating that these models are much better than models that make random predictions. This ROC plot shows that the DenseNet201 model, trained using preprocessing techniques such as CLAHE, performs excellent classification for detecting diseases on grape leaves. An AUC of 1.00 for all classes indicates that the model can accurately discriminate between different conditions with minimal error.

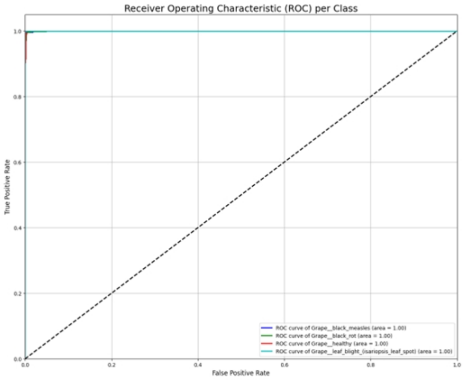

Figure 8. ROC curve of the model

3.2. Discussion

According to the Efficacy model, this model is very effective and can be used with high confidence to diagnose grape leaf diseases. For generalization, a high level of accuracy indicates that the model has good generalization capabilities, which means that this model will work well on data that has never been seen before. Overall, the model effectively classified grape leaf diseases, with excellent performance in all classes. The low error rate indicates that this model is reliable for practical applications in detecting grape leaf diseases in the field. Combining the results of this confusion matrix with other metrics, such as the classification report, ROC curve, and model loss/accuracy, provides a comprehensive picture of the superiority and efficiency of the model in this classification task.

4. CONCLUSION

This research presents a grape leaf disease classification model using the CLAHE technique and DenseNet201 architecture. Based on the model performance evaluation, the overall accuracy and performance of the model were obtained, which reached an overall accuracy of 99.61%, which shows excellent performance in grape leaf disease classification. The recall, precision, and F1 scores for all classes are also very high, with the highest F1 score for the ”leaf blight” class (1.0000) and the lowest for the ”healthy” class (0.9634). ROC curve: The ROC curve shows that all classes have an area under the curve (AUC) of 1.00, which illustrates perfect discrimination between different classes. This shows that the model effectively separates positive and negative cases. Loss and Accuracy Graph: The Loss and Accuracy graph shows a good learning trend during the training process. The loss values decreased, and the accuracy increased consistently with the training and validation data, with no signs of overfitting. The confusion matrix results also show that the model has a very low error rate in classifying the four classes of grape leaf diseases. Most instances were correctly classified, with only a few instances misclassified, further confirming the high accuracy of this model. The overall evaluation of the model shows that the developed grape leaf disease classification is highly effective and reliable for identifying different types of grape leaf diseases. The following steps can be taken for further improvement, including expanding the data set by collecting more data from different conditions and grape varieties to boost the model’s generalization capability. To enhance the model’s ability to handle variations in the input image, implement additional data augmentation techniques. A thorough hyperparameter search will also be conducted to determine the optimal model configuration.

5. ACKNOWLEDGEMENTS

The Acknowledgments section is optional. Research sources can be included in this section.

6. DECLARATIONS

AUTHOR CONTIBUTION

Rudi Kurniawan conducted the research, analyzed data, created the solution, and wrote the paper. Lukman Sunardi analyzes and helps correct each model’s results.

FUNDING STATEMENT

This research received no financial support from any institution or organization. The authors fully bore all research and analysis costs.

COMPETING INTEREST

The authors declare that there are no conflicts of interest related to this research. All research activities, data analysis, and article preparation were performed independently without financial support or affiliation that could influence the research results.

REFERENCES

[1] S. Mohamad, A. Banakar, K. Asefpour, and Y. Ampatzidis, "Diagnosis of grape leaf diseases using automatic K -means clustering and machine learning," *Smart Agricultural Technology*, vol. 3, no. March 2022, p. 100081, 2023, https://doi.org/10.1016/j.atech.2022.100081.

[2] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, and A. Siddiqua, "Plant leaf disease detection using computer vision and machine learning algorithms," *Global Transitions Proceedings journal*, vol. 3, no. April, pp. 305–310, 2022, https://doi.org/10.1016/j.gltp.2022.03.016.

[3] P. V. Yeswanth, S. Deivalakshmi, S. George, and S.-B. Ko, "Residual Skip Network-Based Super-Resolution for Leaf Disease Detection of Grape Plant," *Circuits, Systems, and Signal Processing*, vol. 42, no. 11, pp. 6871–6899, 2023, https://doi.org/10.1007/s00034-023-02430-2.

[4] Z. Huang, A. Qin, J. Lu, A. Menon, and J. Gao, "Grape Leaf Disease Detection and Classification Using Machine Learning," in *IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom) Green Computing and Communications (GreenCom)*, 2020, pp. 870–877, https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00150.

[5] Y. Peng, S. Zhao, and J. Liu, "Fused-Deep-Features Based Grape Leaf Disease Diagnosis," *Agronomy*, vol. 11, no. 11, pp. 1–17, 2021, https://doi.org/10.3390/agronomy11112234.

[6] R. Dwivedi, S. Dey, C. Chakraborty, and S. Tiwari, "Grape Disease Detection Network based on Multi-task Learning and Attention Features," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 1–8, 2021, https://doi.org/10.1109/JSEN.2021.3064060.

[7] R. Nagi and S. S. Tripathy, "Deep convolutional neural network based disease identification in grapevine leaf images," *Multimedia Tools and Applications*, vol. 81, no. 18, pp. 24 995–25 006, 2022, https://doi.org/10.1007/s11042-022-12662-0.

[8] R. M. Math and N. V. Dharwadkar, "Early detection and identification of grape diseases using convolutional neural networks," *Journal of Plant Diseases and Protection*, vol. 129, no. 3, pp. 521–532, 2022, https://doi.org/10.1007/s41348-022-00589-5.

[9] M. Shantkumari and S. V. Uma, "Grape leaf segmentation for disease identification through adaptive Snake algorithm model," *Multimedia Tools and Applications*, vol. 80, November, pp. 8861–8879, 2020, https://doi.org/10.1007/s11042-020-09853-y.

[10] Z. Zinonos, S. Gkelios, and A. L. A. F. Khalifeh, "Grape Leaf Diseases Identification System Using Convolutional Neural Networks and LoRa Technology," *IEEE Access*, vol. 10, Desember, pp. 122–133, 2022, https://doi.org/10.1109/ACCESS.2021.3138050.

[11] V. K. Vishnoi, K. Kumar, and B. Kumar, "Plant disease detection using computational intelligence and image processing," *Journal of Plant Diseases and Protection*, vol. 128, no. 1, pp. 19–53, 2021, https://doi.org/10.1007/s41348-020-00368-0.

[12] J. Zhu and A. Wu, "Identification of grape diseases using image analysis and BP neural networks," *Multimedia Tools and Applications*, vol. 79, January, pp. 14 539–14 551, 2019, https://doi.org/10.1007/s11042-018-7092-0.

[13] B. I. N. Liu, C. Tan, S. Li, J. He, and H. Wang, "A Data Augmentation Method Based on Generative Adversarial Networks for Grape Leaf Disease Identification," *IEEE Access*, vol. 8, June, pp. 102 188–102 198, 2020, https://doi.org/10.1109/ACCESS.2020.2998839.

[14] M. Ji, L. Zhang, and Q. Wu, “Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks,” *Information Processing in Agriculture*, vol. 7, no. 3, pp. 418–426, 2020, https://doi.org/10.1016/j.inpa.2019.10.003.

[15] S. Lauguico and E. Dadios, “Grape Leaf Multi-disease Detection with Confidence Value Using Transfer Learning Integrated to Regions with Convolutional Neural Networks,” in *2020 IEEE Region 10 Conference (Tencon)*, 2020, pp. 767–772, https://doi.org/10.1109/TENCON50793.2020.9293866.

[16] C. Zhou, Z. Zhang, S. Zhou, J. Xing, Q. Wu, and J. I. A. Song, “Grape Leaf Spot Identification Under Limited Samples by Fine Grained-GAN,” *IEEE Access*, vol. 9, pp. 100 480–100 489, 2021, https://doi.org/10.1109/ACCESS.2021.3097050.

[17] W. Guo, Q. Feng, X. Li, S. Yang, and J. Yang, “Grape leaf disease detection based on attention mechanisms,” *Int J Agric & Biol Eng*, vol. 15, no. 5, pp. 205–212, 2022, https://doi.org/10.25165/j.ijabe.20221505.7548.

[18] H. Jin, Y. Li, J. Qi, J. Feng, D. Tian, and W. Mu, “GrapeGAN: Unsupervised image enhancement for improved grape leaf disease recognition,” *Computers and Electronics in Agriculture*, vol. 198, July, p. 107055, 2022, https://doi.org/10.1016/j.compag.2022.107055.

[19] J. Lin, X. Chen, R. Pan, T. Cao, J. Cai, Y. Chen, X. Peng, T. Cernava, and X. Zhang, “GrapeNet : A Lightweight Convolutional Neural Network Model for Identification of Grape Leaf Diseases,” *Agriculture (Switzerland)*, vol. 12, no. 6, 2022, https://doi.org/10.3390/agriculture12060887.

[20] S. P. Praveen, R. Nakka, A. Chokka, V. N. Thatha, S. S. Vellela, and U. Sirisha, “A Novel Classification Approach for Grape Leaf Disease Detection Based on Different Attention Deep Learning Techniques,” *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, pp. 1199–1209, 2023, https://doi.org/10.14569/IJACSA.2023.01406128.

[21] T. Lu, B. Han, L. Chen, F. Yu, and C. Xue, “A generic intelligent tomato classification system for practical applications using DenseNet-201 with transfer learning,” *Scientific Reports*, pp. 1–8, 2021, https://doi.org/10.1038/s41598-021-95218-w.

[22] N. Paul, G. C. Sunil, D. Horvath, and X. Sun, “Deep learning for plant stress detection : A comprehensive review of technologies , challenges , and future directions,” *Computers and Electronics in Agriculture*, vol. 229, no. July 2024, p. 109734, 2025, https://doi.org/10.1016/j.compag.2024.109734.

[23] G. Geetharamani and A. P. J, “Identification of plant leaf diseases using a nine-layer deep convolutional neural network,” *Computers and Electrical Engineering*, vol. 76, pp. 323–338, 2019, https://doi.org/10.1016/j.compeleceng.2019.04.011.

[24] C. C. Ukwuoma, Z. Qin, B. Bin, F. Akhtar, O. Bamisile, A. Y. Muaad, D. Addo, and M. A. Al-antari, “A Hybrid Explainable Ensemble Transformer Encoder for Pneumonia Identification from Chest X-ray Images,” *Journal of Advanced Research*, vol. 48, pp. 191–211, 2023, https://doi.org/10.1016/j.jare.2022.08.021.

[25] C. C. Ukwuoma, D. Cai, B. Bin, O. Bamisile, H. Adun, Z. Al-huda, and M. A. Al-antari, “Deep Learning Framework for Rapid and Accurate Respiratory COVID-19 Prediction using Chest X-ray Images,” *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 7, p. 101596, 2023, https://doi.org/10.1016/j.jksuci.2023.101596.

[26] P. Härtinger and C. Steger, “Adaptive histogram equalization in constant time,” *Journal of Real-Time Image Processing*, vol. 21, no. 3, p. 93, 2024, https://doi.org/10.1007/s11554-024-01465-1.

[27] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *Journal of Big Data*, vol. 6, no. 1, 2019, https://doi.org/10.1186/s40537-019-0197-0.

[28] G. Huang, Z. Liu, and L. V. D. Maaten, “Densely Connected Convolutional Networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, https://doi.org/10.48550/arXiv.1608.06993.

[29] L. S. Bernardo, R. Damaševičius, V. H. C. De Albuquerque, and R. Maskeliūnas, “A Hybrid Two-stage SqueezeNet and Support Vector Machine System for Parkinson’s Disease Detection based on Handwritten Spiral Patterns,” *International Journal of Applied Mathematics and Computer Science*, vol. 31, no. 4, pp. 549–561, 2021, https://doi.org/10.34768/amcs-2021-0037.

[30] P. B. Le, “ROC Curves , Loss Functions , and Distorted Probabilities in Binary Classification,” *Mathematics*, vol. 10, no. 9, pp. 1–13, 2022, https://doi.org/10.3390/math10091410.

[31] M. P. Niroomand, C. T. Cafolla, J. W. R. Morgan, and D. J. Wales, “Characterising the area under the curve loss function landscape,” *Machine learning science and technology*, vol. 3, no. 1, pp. 1–15, 2022, https://doi.org/10.1088/2632-2153/ac49a9.