Bulletin of Computer Science and Electrical Engineering
Vol. 4, No. 2, December 2023, pp. 83~88
ISSN: 2722-7324, DOI: 10.25008/bcsee.v4i2.1184

Enhancing Rose Leaf Disease Detection Accuracy Using Optimized CNN Parameters

Latifah Nabilah[1], Nisar[2], Amnah[3], Septilia Arfida[4]
[1,4]Department of Computer Science, Institute of Informatika & Business Darmajaya, Indonesia

Article Info

Article history:

Received Oct 10, 2023
Revised Dec 09, 2023
Accepted Dec 30, 2023

Keywords:

CNN model
Rose leaf disease detection
Optimization
RMSProp optimizer
Early stopping

ABSTRACT

The CNN model developed in this study demonstrated remarkable performance, achieving an outstanding validation accuracy of 99.96%. Through experimentation, it was found that employing the RMSprop optimizer with a learning rate of 0.001 yielded superior results compared to the Adam optimizer utilized in previous iterations. Additionally, increasing the number of epochs from 10 to 20 resulted in a significant enhancement in accuracy, highlighting the importance of iterative training for model refinement. Moreover, the implementation of Early Stopping proved to be a valuable technique, effectively conserving training time by halting the training process once optimal accuracy levels were reached. These findings underscore the efficacy of various optimization strategies in bolstering the performance of CNN models for rose leaf disease detection. The achieved accuracy rates signify a substantial advancement in disease detection technology, holding promise for enhancing agricultural productivity and ensuring plant quality. This research contributes valuable insights into the optimization of CNN parameters, paving the way for further advancements in automated disease detection systems in the field of agriculture.

Corresponding Author:

Latifah Nabilah,
Department of Computer Science,
Institute of Informatika & Business Darmajaya,
93 Z.A. Pagar Alam Street, Kedaton, Bandar Lampung City, Lampung, Indonesia, 34152.
Email: latifah.2011010045@mail.darmajaya.ac.id

1. INTRODUCTION

The global production and export of roses have seen significant economic importance, with the export value surpassing 175 million USD in 2019, primarily led by top exporting countries such as the Netherlands, Denmark, Uganda, Germany, and Canada [1]. In Indonesia, according to the Central Bureau of Statistics, the production of rose cut stems increased by 169.106 million in 2022 [2]. Roses have long been cherished for their enchanting aroma and iconic symbolism of love and elegance, adorning gardens, floral arrangements, and special occasions worldwide. However, like any other plant, roses are susceptible to leaf diseases, which can jeopardize plant health and reduce flower quality. Rose leaf diseases have become a serious concern for farmers and rose producers worldwide[3]. Leaf diseases such as rose slugs, rust, and sawfly larvae pose significant threats to the health and productivity of rose plants [4].

Manual diagnosis of rose leaf diseases requires significant skills and experience, while the process is also time-consuming and costly [1] Previous research has attempted to address this issue, but there is still room for further improvement. Thus, in this study, we propose a novel approach to enhance the efficiency and accuracy of diagnosing rose leaf diseases using state-of-the-art technology[5]. We believe that the proposed approach will provide significant innovative value in efforts to protect rose plants from disease outbreaks and improve the well-being of farmers and the overall sustainability of the floriculture sector.

In the previous research conducted Basak [4], an accuracy of 95.65% was achieved. This indicates that the approach used in that study yielded satisfactory results in diagnosing leaf diseases in rose plants. However, there is still room for further improvement in accuracy and efficiency of the diagnostic process. Therefore, this study aims to propose a new approach that can enhance diagnostic accuracy and address potential limitations of the previous approach.

This research is limited to the utilization of rose leaf image data, encompassing several common categories of leaf diseases. The development of the model will focus on the detection and classification of rose leaf diseases based on digital images with a high level of accuracy. The aim of this study is to enhance the accuracy of rose leaf disease diagnosis through the optimization of convolutional neural network models, taking into consideration factors such as activation functions, optimizers, and the number of epochs.

2. RESEARCH METHOD

The dataset utilized in this study was sourced from kaggle, a popular platform for accessing and sharing datasets. Kaggle provides a diverse range of datasets contributed by researchers and data enthusiasts worldwide. In this research, we accessed a dataset specifically curated for rose leaf images, containing various instances of leaf diseases commonly found in rose plants. The dataset from kaggle served as the primary source of image data for training and evaluating the convolutional neural network models developed in this study. The procedures have been carried out:

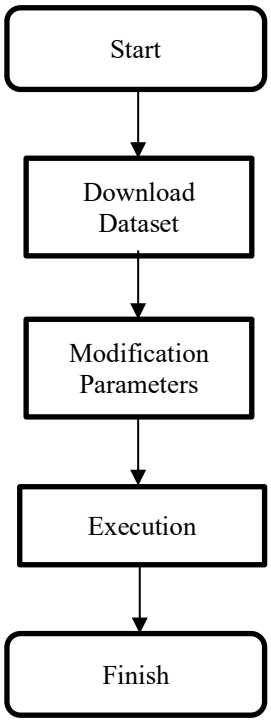


Figure 1 Research Workflow Flowchart

2.1. Dataset

The images of rose leaves were obtained from the Kaggle platform[6], which provides a dataset consisting of a total of 14,910 images. This dataset is divided into three main categories: Healthy Leaf Rose, Rose Rust, and Rose Sawfly/Rose Slug. Each category contains images representing the respective conditions with various variations[7]. The data collection process involved downloading the dataset from the Kaggle platform and directly storing it in Google Colab.

2.2. Modification the activation function

After conducting preliminary analysis, we tested the use of Leaky ReLU activation function in both convolutional and dense layers of the CNN model [8]. However, following experimentation, we found that the implementation of Leaky ReLU did not yield a significant improvement in our model's accuracy. Consequently, we decided to revert to using the ReLU (Rectified Linear Unit) activation function, which was initially employed in the model [9].

This decision was based on evaluation results indicating that the utilization of Leaky ReLU did not deliver the expected performance enhancement in our model. Despite Leaky ReLU being designed to address the issue of "dead neurons" that may occur with ReLU, in our case, the change did not result in significant improvement in feature representation or acceleration of convergence during the training process [10].

Thus, we opted to reinstate the ReLU activation function for both convolutional and dense layers in our model. This modification was implemented by reverting the activation parameter in these layers to 'relu' [11]. Subsequently, we recompiled the model using the same optimizer, namely Adam, and categorical crossentropy loss function as before. This decision was made based on the evaluation of experimental results, which indicated that the use of ReLU offered better performance within the context of our model.

2.3. Modification optimizer

After changing the activation function, the next step we took was to switch the optimizer from Adam to RMSprop[12]. Additionally, we introduced additional settings, including setting the learning rate to 0.001 for RMSprop[13], and implemented the early stopping method to improve model training[14]. Setting the learning rate to 0.001 was applied to adjust the learning rate in RMSprop, allowing the model to make finer weight adjustments during training. Furthermore, we incorporated the early stopping method to prevent overfitting and expedite convergence[15]. The early stopping method monitored the model's performance on validation data during training and halted training when the performance did not improve for a certain number of consecutive epochs[16].

2.4. Modification of the Number of Epochs

As the final step in fine-tuning the CNN model, we decided to increase the number of epochs from 10 to 20. This addition of epochs was made considering the monitoring results obtained through the early stopping technique. Early stopping was implemented to monitor the model's performance on the validation dataset. By adding epochs, we provided the model with more iterations to learn complex patterns from the training data[17]. However, we remained vigilant about the potential for overfitting by utilizing the early stopping mechanism to halt training if there was no significant improvement in validation loss. This change was expected to give our model more time to converge and improve the accuracy of rose leaf disease detection. The implementation of this modification reflects our efforts to adjust the model training process to achieve optimal results.

3. RESULTS AND DISCUSSION

Table 1. Experimental Results at Epoch 10

Epoch	Activation	Optimezer	Learning Rate	Early Stoping	Accuracy
10	ReLU	Adam			95,65
10	Leaky ReLU	Adam			97,25
10	ReLU	RMSProp	0,001		99,96
10	Leaky ReLU	RMSProp	0,001		98,98

In the development experiment of the Convolutional Neural Network (CNN) model for diagnosing rose leaf diseases, we conducted variations in key parameters such as activation function, optimizer, and learning rate, aiming to enhance the model's accuracy. In the first trial, we utilized the ReLU activation function and Adam optimizer, achieving an accuracy of 95.65%. The utilization of the Leaky ReLU activation function in the second trial boosted the accuracy to 97.25%. The third trial demonstrated that the combination of ReLU with RMSProp optimizer and a learning rate of 0.001 yielded very high accuracy, reaching 99.96%. However, in the fourth trial, despite retaining Leaky ReLU, employing the RMSProp optimizer resulted in slightly lower accuracy, namely 98.98%. These outcomes highlight the significance of selecting the optimal combination of model parameters to attain the desired performance.

Table 2. Experimental Results at Epoch 20

Epoch	Activation	Optimezer	Learning Rate	Early Stoping	Accuracy
20	ReLU	Adam	0,001	Epoch 17	97,16
20	Leaky ReLU	Adam	0,001		97,33
20	ReLU	RMSProp	0,001	Epoch 10	98,50
20	Leaky ReLU	RMSProp	0,001	Epoch 13	99,47

In this experiment, tests were conducted using a Convolutional Neural Network (CNN) model with variations in the number of epochs, types of activation functions (ReLU and Leaky ReLU), and the use of optimizers (Adam and RMSProp). In each test, the learning rate parameter was set to 0.001. The results of the experiment showed that in the model using ReLU activation function and Adam optimizer, training was stopped at Epoch 17 with an accuracy of 97.16%. Meanwhile, the same model using Leaky ReLU activation

function achieved an accuracy of 97.33% without early stopping notification. Furthermore, in the model using RMSProp optimizer and ReLU activation function, training was stopped at Epoch 10 with an accuracy of 98.50%. On the other hand, in the same model using Leaky ReLU activation function, training was stopped at Epoch 13 with an accuracy reaching 99.47%.

It is worth noting that early stopping was not applied in the second set of experiments to evaluate the effectiveness of early stopping in improving model convergence.

3.1. The best epoch results

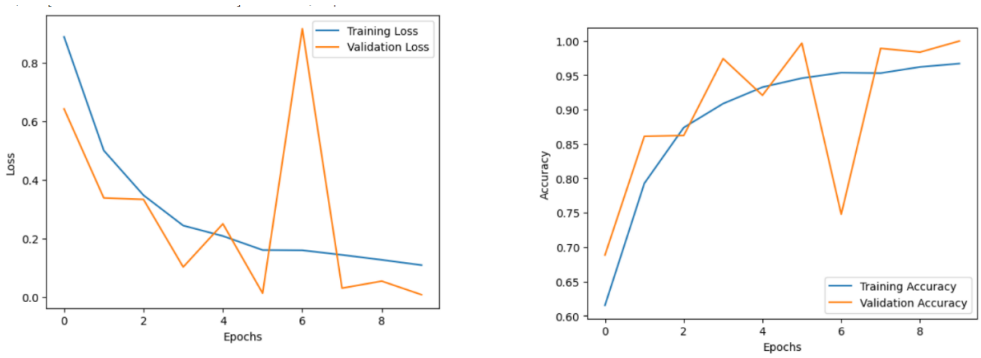
```
Found 7350 images belonging to 3 classes.
Found 7350 images belonging to 3 classes.
Epoch 1/10
230/230 [==============================] - 1346s 6s/step - loss: 0.8887 - accuracy: 0.6152 - val_loss: 0.6424 - val_accuracy: 0.6883
Epoch 2/10
230/230 [==============================] - 1376s 6s/step - loss: 0.5003 - accuracy: 0.7927 - val_loss: 0.3379 - val_accuracy: 0.8611
Epoch 3/10
230/230 [==============================] - 1382s 6s/step - loss: 0.3472 - accuracy: 0.8737 - val_loss: 0.3329 - val_accuracy: 0.8622
Epoch 4/10
230/230 [==============================] - 1312s 6s/step - loss: 0.2435 - accuracy: 0.9086 - val_loss: 0.1016 - val_accuracy: 0.9740
Epoch 5/10
230/230 [==============================] - 1388s 6s/step - loss: 0.2077 - accuracy: 0.9325 - val_loss: 0.2495 - val_accuracy: 0.9207
Epoch 6/10
230/230 [==============================] - 1316s 6s/step - loss: 0.1598 - accuracy: 0.9456 - val_loss: 0.0120 - val_accuracy: 0.9966
Epoch 7/10
230/230 [==============================] - 1374s 6s/step - loss: 0.1589 - accuracy: 0.9536 - val_loss: 0.9168 - val_accuracy: 0.7476
Epoch 8/10
230/230 [==============================] - 1350s 6s/step - loss: 0.1432 - accuracy: 0.9529 - val_loss: 0.0294 - val_accuracy: 0.9890
Epoch 9/10
230/230 [==============================] - 1340s 6s/step - loss: 0.1262 - accuracy: 0.9619 - val_loss: 0.0534 - val_accuracy: 0.9834
Epoch 10/10
230/230 [==============================] - 1390s 6s/step - loss: 0.1082 - accuracy: 0.9668 - val_loss: 0.0070 - val_accuracy: 0.9996
```

Figure 2 The best epoch results

The training results of the Convolutional Neural Network (CNN) model on a dataset consisting of 3 classes with a total of 7350 images showed significant improvement over 10 epochs. Initially, the model achieved around 61.52% accuracy in the first epoch. However, as the number of epochs increased, the model's performance improved consistently. By the end of training, the model achieved very high accuracy levels, reaching approximately 96.68% on the training data and 99.96% on the validation data.

Throughout the training process, there was a consistent increase in accuracy and a significant decrease in loss value. This indicates that the model successfully learned complex patterns in the dataset and performed classification very well. The final results of this experiment demonstrate that the CNN model I developed is capable of producing highly accurate predictions for the classes in the given dataset.

3.2. Graphical Results



(a) (b)

Figure 3 Graphical Results

Image (a) shows a chart with two lines representing different types of loss, presumably from a machine learning model training process. There are two lines on the chart, one is blue and labeled "Training Loss" and the other is orange and labeled "Validation Loss." The x-axis of the chart is labeled "Epochs," which generally refers to complete passes through a training dataset during the machine learning process. The scale of the x-axis runs from 0 to 8, suggesting that the data covers 9 epochs. The y-axis is labeled "Loss," which is a metric used to evaluate how well the model's predictions match the actual data. The scale on the y-axis runs from 0 to 1, indicating that the loss values fall within this range. Both lines show fluctuations in loss value, with the training loss starting high at around 0.8 and decreasing to about 0.2 by the 8th epoch. The validation loss starts around 0.6, increases to about 0.7, has sharp declines and spikes around the 3rd and 4th

epoch, and ends up approximately at 0.4 by the 8th epoch. The behavior of these lines could indicate how well the model is learning from the data. Generally, the desired outcome is for both training and validation loss to decrease over time, and ideally, for the validation loss to be low and stable, indicating good generalization of the learned model to new data. The spikes and fluctuations seen in the validation loss curve suggest that the model may be experiencing some instability during training, such as overfitting to the training data or underfitting to the validation data at certain epochs.

Image (b) is a line graph plotting "Accuracy" against "Epochs." The Accuracy is on the y-axis and ranges from 0.65 to 1.00, while Epochs are on the x-axis and range from 0 to 9 (perhaps representing training iterations in a machine learning context). There are two lines on the graph: one labeled "Training Accuracy" and the other "Validation Accuracy." The "Training Accuracy" line is in blue and generally trends upwards as the number of epochs increases, indicating an improvement in accuracy over time on the training dataset. The "Validation Accuracy" line is in orange, and it shows more variability. It increases, then drops at the third epoch, rises again, and falls slightly before the last epoch. This could suggest that while the model is learning, it may be experiencing some overfitting or instability in generalizing to new, unseen data. Overall, the graph is demonstrating a typical scenario in the training of machine learning models where the model's performance on training data improves while the performance on validation data may experience fluctuations.

3.3. Confusion Matrix

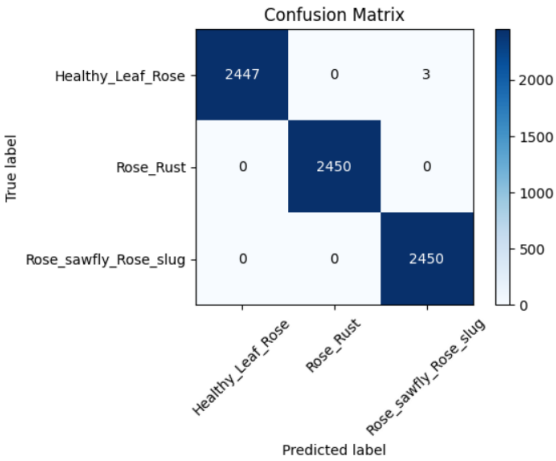


Figure 4 Confusion Matrix

The experimental findings unveil the remarkable efficacy of the developed model, which achieved an exceptional accuracy rate of 99.967%. This outstanding performance underscores the model's adeptness in accurately discerning and categorizing different disease classes. Delving deeper into the assessment, the confusion matrix offers a detailed breakdown of the model's predictions. In the case of "Healthy_Leaf_Rose," the model accurately identified 2447 instances, while misclassifying 3 cases. Similarly, for "Rose_Rust," the model made precise predictions in all 2450 cases without any misclassifications. Moreover, in the category of "Rose_sawfly_Rose_slug," the model achieved perfect accuracy, correctly classifying all 2450 instances. Such meticulous evaluation through the confusion matrix provides invaluable insights into the model's performance, elucidating both correct classifications (true positives) and misclassifications (false positives and false negatives) across various disease classes.

3.4. Classification Report

```
Found 7350 images belonging to 3 classes.
230/230 [==============================] - 341s 1s/step
Confusion Matrix:
[[2447    0    3]
 [   0 2450    0]
 [   0    0 2450]]
Classification Report:
                        precision    recall  f1-score   support

    Healthy_Leaf_Rose       1.00      1.00      1.00      2450
            Rose_Rust       1.00      1.00      1.00      2450
Rose_sawfly_Rose_slug       1.00      1.00      1.00      2450
```

Figure 5 Classification Report

The model has very high precision, recall, and f1-score values for each class, indicating excellent performance. Accuracy also reached 1.00 which shows that the model performs predictions very well overall.

4. CONCLUSION

In conclusion, our study aimed to improve the accuracy of rose leaf disease detection through CNN model optimization and parameter adjustments. We expected enhanced model performance through various modifications including activation function tuning, optimizer selection, and adjusting the number of epochs. Experimental results revealed that utilizing ReLU activation function and RMSprop optimizer over 10 epochs yielded the highest accuracy of 99.96%. Moving forward, there are prospects for further research to explore additional optimization techniques and real-world application of the optimized model for rose leaf disease detection. This study provides a foundation for future research in plant disease detection, with the potential to enhance agricultural practices and disease management.

REFERENCES

[1] S. Nuanmeesri, "A hybrid deep learning and optimized machine learning approach for rose leaf disease classification," *Eng. Technol. &Applied Sci. Res.*, 2021, [Online]. Available: https://www.etasr.com/index.php/ETASR/article/view/4455

[2] "Produksi Tanaman Florikultura (Hias)."

[3] M. Ali-Al-Alvy, G. K. Khan, M. J. Alam, S. Islam, M. Rahman, and M. S. Rahman, "Rose Plant Disease Detection using Deep Learning," *7th Int. Conf. Trends Electron. Informatics, ICOEI 2023 - Proc.*, no. April, pp. 1244–1249, 2023, doi: 10.1109/ICOEI56765.2023.10126031.

[4] S. K. Basak, "Unveiling the Enigma : Advancing Rose Leaf Disease Detection with Transformed Images and Convolutional Neural Networks," *Zenodo*, vol. 1, pp. 1–7, 2023.

[5] A. Rajbongshi, T. Sarker, M. M. Ahamad, and M. M. Rahman, "Rose Diseases Recognition using MobileNet," *4th Int. Symp. Multidiscip. Stud. Innov. Technol. ISMSIT 2020 - Proc.*, 2020, doi: 10.1109/ISMSIT50672.2020.9254420.

[6] *et al.*, "Plant Leaf Disease Detection and Classification Using Deep Learning: A Review and A Proposed System on Bangladesh's Perspective," *Int. J. Sci. Bus.*, vol. 28, no. 1, pp. 193–204, 2023, doi: 10.58970/ijsb.2214.

[7] Xiaoguang Lu, Yunhong Wang, and A. K. Jain, "Combining classifiers for face recognition," no. March, pp. III–13, 2004, doi: 10.1109/icme.2003.1221236.

[8] A. Mujhid, S. Surono, N. Irsalinda, and A. Thobirin, "Comparison and Combination of Leaky ReLU and ReLU Activation Function and Three Optimizers on Deep CNN for COVID-19 Detection," *Front. Artif. Intell. Appl.*, vol. 358, pp. 50–57, 2022, doi: 10.3233/FAIA220369.

[9] Y. Bai, "RELU-Function and Derived Function Review," *SHS Web Conf.*, vol. 144, p. 02006, 2022, doi: 10.1051/shsconf/202214402006.

[10] F. Schilling, "The Effect of Batch Normalization on Deep Convolutional Neural Networks," p. 113, 2016, [Online]. Available: http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A955562&dswid=-5716

[11] D. Bertoin, J. Bolte, S. Gerchinovitz, and E. Pauwels, "Numerical influence of ReLU'(0) on backpropagation," *Adv. Neural Inf. Process. Syst.*, vol. 1, no. 0, pp. 468–479, 2021.

[12] P. Meißner, H. Watschke, J. Winter, and T. Vietor, "Artificial neural networks-based material parameter identification for numerical simulations of additively manufactured parts by material extrusion," *Polymers (Basel).*, vol. 12, no. 12, pp. 1–28, 2020, doi: 10.3390/polym12122949.

[13] T. Nagaoka, "Hyperparameter optimization for deep learning-based automatic melanoma diagnosis system," *Adv. Biomed. Eng.*, vol. 9, pp. 225–232, 2020, doi: 10.14326/abe.9.225.

[14] K. Nakamura, B. Derbel, K. J. Won, and B. W. Hong, "Learning-rate annealing methods for deep neural networks," *Electron.*, vol. 10, no. 16, pp. 1–12, 2021, doi: 10.3390/electronics10162029.

[15] M. Alalhareth and S. C. Hong, "An Adaptive Intrusion Detection System in the Internet of Medical Things Using Fuzzy-Based Learning," *Sensors (Basel).*, vol. 23, no. 22, 2023, doi: 10.3390/s23229247.

[16] T. Miseta, A. Fodor, and Á. Vathy-Fogarassy, "Surpassing early stopping: A novel correlation-based stopping criterion for neural networks," *Neurocomputing*, vol. 567, no. November 2023, p. 127028, 2024, doi: 10.1016/j.neucom.2023.127028.

[17] S. F. Ahmed *et al.*, *Deep learning modelling techniques: current progress, applications, advantages, and challenges*, vol. 56, no. 11. Springer Netherlands, 2023. doi: 10.1007/s10462-023-10466-8.