Comparative Analysis of Classical Machine Learning Models and Deep Learning in Rose Leaf Disease Classification

1st Kent Amadeo Timotheus
Computer Science Department
School of Computer Science
Bina Nusantara University
Bekasi Regency, Indonesia
kent.timotheus@binus.ac.id

2nd Theodore Zachary
Computer Science Department
School of Computer Science
Bina Nusantara University
Tangerang, Indonesia
theodore.zachary@binus.ac.id

3rd Jason Ho
Computer Science Department
School of Computer Science
Bina Nusantara University
Tangerang, Indonesia
jason.ho@binus.ac.id

4th Diana
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
diana@binus.edu

5th Shania Priccilia
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
shania.priccilia@binus.ac.id

***Abstract*—This paper presents a comparative analysis of classical machine learning (ML) models and deep learning (DL) models for the classification of rose leaf diseases. The study evaluated three ML models (*k*-Nearest Neighbors, Random Forest, and Support Vector Machine) and three DL models (Convolutional Neural Network, ResNet-50, and VGG16Net) to classify four categories of rose leaf diseases: healthy leaves, black spot, leaf holes, and dry leaves. An initial dataset of 3,113 images was augmented to 12,000 images to ensure class balance. Image preprocessing involved region segmentation, cropping, and resizing to 224×224 pixels. For the ML models, features were extracted using Grey-Level Co-occurrence Matrix (GLCM), specifically Contrast, Dissimilarity, Homogeneity, Energy, and Correlation. The dataset was divided into a 70:30 train-validation ratio. The results indicate that two of the deep learning models, CNN and VGG16Net, significantly surpassed the ML models in accuracy, with CNN achieving the highest accuracy at 98.87% and VGG16Net at 97.58%. While the ML models performed consistently, Random Forest achieved a competitive accuracy of 80.88%. Notably, ResNet-50 underperformed all other models.**

Keywords*—*machine learning, deep learning, rose leaf diseases, neural networks, random forests

I. Introduction

The rose plant is cultivated in various countries such as India, China, and numerous other countries. Acknowledged globally as 'the queen of flowers,' roses are considered a high-value flower [1]. The global demand for roses remains substantial due to their widespread application in numerous industries, including perfumery, cosmetics, pharmaceuticals, and related sectors, thereby contributing to their significant economic value [2], [3]. In 2019, it was a commodity for countries such as the Netherlands, Uganda, and Denmark, making the cultivation of rose plants of significant agricultural importance [4]. In cultivating roses, there are challenges because roses are susceptible to diseases. Differences in location, climate, or environmental conditions can contribute to the development of diseases [5]. Such diseases often result in a decline in quality and overall value, with detrimental impacts on both individual growers and the broader agricultural economy [5]. The leaves of a rose plant can be an indicator of a disease, with bacteria as one of the sources of the disease in roses [6]. However, accurate visual identification of rose leaf diseases is labor-intensive and requires significant human expertise. Therefore, computer vision applied to rose leaf images can serve as a machine-based approach for the preventive detection of rose diseases.

Early detection of rose diseases is essential for minimizing their impact on the health and quality of the flowers [4], [7]. With the rapid advancement of technology, various innovative approaches have been developed for disease detection. Two prominent technologies that can be utilized for this purpose are classical Machine Learning (ML) and Deep Learning (DL), both of which offer effective solutions for identifying and diagnosing plant diseases [8]. Some popular Machine Learning algorithms are *k*-Nearest Neighbors (*k*-NN), Random Forests (RF), and Support Vector Machine (SVM) [9], [10], [11], [12]. Popular Deep Learning models include Convolutional Neural Network (CNN) [9], and Transfer Learning models, VGGNet, ResNet [12]. These models were selected for this study due to their established effectiveness and frequent application in image processing tasks. In this paper, image disease detection is studied using both ML and DL, and then a comparative analysis of the accuracy is done.

Comparative analysis of multiple models constitutes an effective methodological approach for determining the optimal model configuration based on quantifiable performance metrics. This study utilized Accuracy, F1-Score, Precision, Recall, and Confusion Matrix as performance metrics. These findings are anticipated to contribute to the development of an efficient and effective agricultural management system that optimizes both the productivity and economic value of rose cultivation [13].

II. Related Works

Rose diseases are caused by various factors, including fungi, bacteria, and viruses. Computer vision, as a modern technological solution, significantly enhances the effectiveness and efficiency of agricultural practices, particularly in rose cultivation productivity. The accuracy of computer vision systems often surpasses that of manual inspection, which typically demands considerable human labor and extensive resources [5], [11].

Numerous studies have implemented image-based detection methodologies for rose foliage and plant specimens, utilizing diverse computational models and algorithms. Convolutional Neural Networks (CNN) and deep learning frameworks represent the predominant algorithmic approaches adopted in this domain. The dominance of deep learning algorithms over classical machine learning is demonstrated through accuracy metrics, especially when applied to large datasets [6], [8], [14]. A study published in MDPI Applied Sciences reported that Support Vector Machine (SVM) models typically achieved 78% accuracy, whereas deep convolutional neural network (DCNN) algorithms attained accuracies as high as 99% [13]. This highlights a significant discrepancy in performance between ML and DL.

A similar study comparing machine learning and deep learning in image-based plant disease detection shows that Deep Learning using CNN is still the best for image classification with 99.3% accuracy. Machine learning only reaches 91.7% using SVM. While both approaches demonstrate favorable outcomes and high accuracy, machine learning models exhibit a comparatively higher error rate than deep learning models [8]. Another study further corroborates the dominance of deep learning [5]. Using SVM independently yielded 93.8% accuracy, whereas its integration with deep learning through Artificial Neural Network (ANN) resulted in 99% accuracy. Alternative models demonstrated variable performance metrics, with Stochastic Gradient Descent (SGD) achieving 96% accuracy, while AlexNet and GoogleNet architectures both attained 99% accuracy. Notably, Convolutional Neural Network (CNN) implementations across diverse datasets consistently demonstrated near-perfect accuracy rates [5].

III. METHOD

A. Image Dataset Gathering

For this study, rose leaf diseases were identified from a collected image dataset. A dataset was utilized, sourced from Kaggle [15]. The dataset comprises 3,113 unique images of rose leaves, capturing diverse leaf morphologies against an isolated white background. It includes images categorized into four classes: Healthy Leaves, Black Spots, Dry Leaves, and Leaf Holes. Although an augmented version of this dataset was available, the original images were chosen for independent augmentation to ensure control over the augmentation process. The images within the dataset possess resolutions of 2448×3264 pixels for vertically oriented images and 3264×2448 pixels for horizontally oriented images.

TABLE I. ROSE LEAF DISEASE IMAGE CLASSIFICATION

No.	Leaf Disease	Number of Images
1	Healthy (disease-free)	818
2	Black Spot	1288
3	Leaf Holes	683
4	Dry	324

B. Image Pre-processing

The fundamental pre-processing steps performed in this research are outlined in Fig. 1.

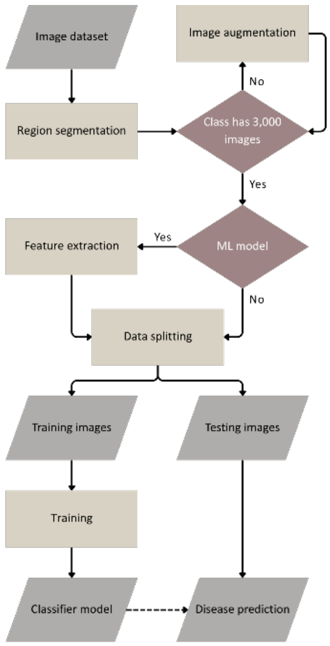

Fig. 1. Flowchart for the basic training and testing steps

1) Image segmentation

Image segmentation was applied to isolate the leaf from its background, thereby emphasizing relevant leaf features and removing extraneous background elements. Segmentation was performed using a threshold mask, which subsequently generated a bounding box through contouring. The thresholding method used in this experiment is the OTSU method [16]. Following thresholding and contouring, images were cropped according to the generated bounding box and resized to 224×224 pixels. An example of the output image is illustrated in Fig. 2.

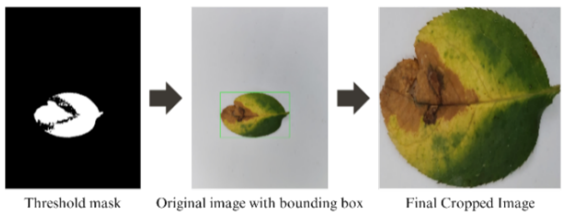

Fig. 2. GrabCut method for image segmentation

2) Image augmentation

The performance of deep learning models is heavily reliant on the volume of training images. A scarcity of images within a specific class can negatively impact the accuracy of the classification. To address this, image augmentation techniques were employed, including vertical and horizontal flips, random rotations (at angles ranging from –90 to 90 degrees), blurring (with a kernel size of up to 3), noise addition (with a standard deviation range of 0.05 to 0.2), and random zoom-in (up to 200%). The probability weights were set to 0.5 for most techniques, apart from random zoom-in (0.75) and random rotations (1.0). During each iteration, only one augmentation method was applied per image. This process significantly increased the number of training images from 3,113 to a balanced total of 12,000, with each class containing 3,000 images, encompassing both original and augmented versions. Examples of the applied augmentation techniques are illustrated in Fig. 3.

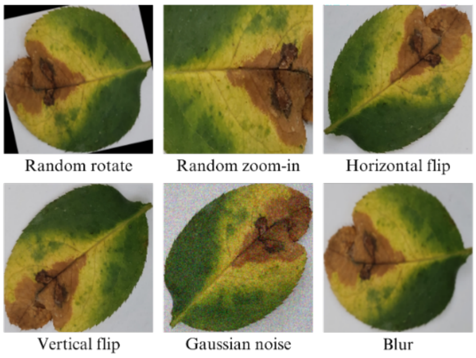

Fig. 3. Image augmentation example

3) Feature extraction

Feature extraction is a critical component of classification processes, providing salient information derived from images [2], [8], [17]. Texture features were extracted using the Grey-Level Co-occurrence Matrix (GLCM), a method commonly employed to analyze image characteristics based on the spatial relationships of pixels. The GLCM extraction yielded five distinct features: Contrast, Dissimilarity, Homogeneity, Energy, and Correlation [7], [8], [17]. The ML models in the experiment then have these features as inputs for classification.

4) Data splitting

Data splitting is done through partitioning the dataset into training and validation sets with a 70:30 ratio. To maintain the original class distribution in both subsets, stratified sampling was also applied based on the disease labels. The models in the experiment were trained using the training data, while the validation data was used to monitor performance and adjust model weights during the training.

C. Machine Learning Modeling

1) K-Nearest Neighbors (k-NN)

k-Nearest Neighbors (*k*-NN) is a supervised machine learning algorithm often used in classification and regression tasks. *k*-NN does not require making assumptions about the data, making it more flexible. This algorithm classifies new data using the majority class from the *k*-nearest neighbor point [8], [17]. A k-NN model is utilized in this experiment, applying GLCM to the preprocessed images to extract extra features, which are then used for input. Through prior testing, the parameter *k* was set to 1 as it resulted in the highest classification accuracy.

2) Random Forest

Random Forest is an ensemble learning machine learning algorithm, capable of solving classification or regression tasks by using multiple decision trees. This suggests a random forest model can handle high-dimensional data and prevent overfitting [2], [12]. GLCM features previously extracted are also utilized in this model. For this experiment, the n_estimators parameter showed optimal model accuracy when its value was set to 200 trees. Random Forest's feature selection capability makes it a suitable model for this research problem.

3) Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm suitable for classification tasks, especially as it is known to be effective in high-dimensional spaces. In this experiment, SVM was utilized with the Radial Basis Function kernel, which is suitable for dealing with non-linear classification problems [8], [12], [17]. To improve generalization, the SVM model used here utilized GLCM extracted features. The penalty parameter C was set to a value of 10, as it suggests good results in improving accuracy and reducing overfitting.

D. Deep Learning Modeling

1) Convolutional Neural Networks (CNN)

The proposed Convolutional Neural Network (CNN) model used in this experiment is a custom sequential architecture consisting of three convolutional blocks and two dense layers. As some models have the input requirement of 224×224 pixels, this CNN model is also designed to be fed an input image format of the same size to be consistent with other models [18]. All convolutional layers use a kernel size of 3×3 and incorporate the Rectified Linear Unit (ReLU) activation function. Before the output of the convolutional blocks enters the dense layers, it also goes through a flatten layer. The last dense layer features a softmax activation function to produce probabilities for the four output classes. Thus, the model has a total of 44,398,148 trainable parameters. A detailed visualization of the architecture is shown in Fig. 4.

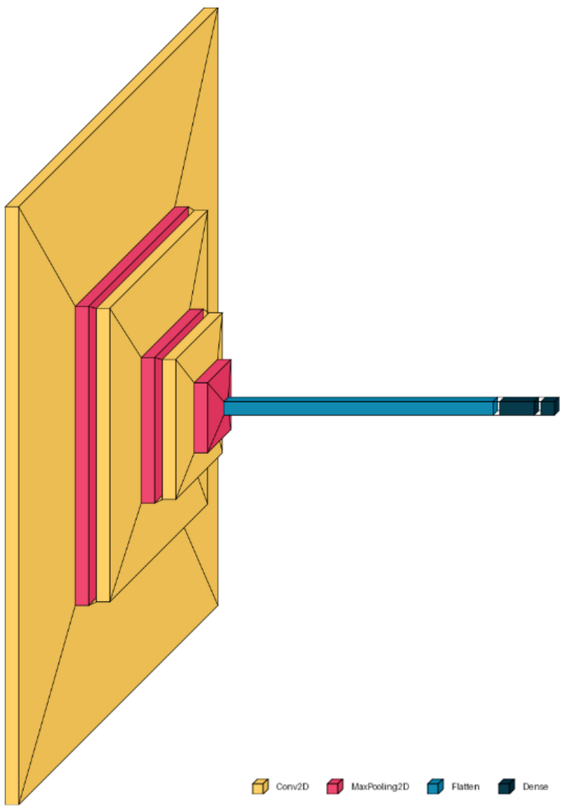

Fig. 4. Proposed CNN architecture

The model was compiled using the Adam optimizer and categorical cross-entropy loss. Training was conducted with a batch size of 32, a learning rate of 0.001, for a total of 10 epochs.

2) ResNet-50

ResNet-50 is a 50-layer deep variant of the Residual Networks (ResNet) architecture family, as introduced in [19]. In this experiment, the ResNet-50 model utilized frozen ImageNet pre-trained base layers, which were combined with a custom classification head. This custom classification head consisted of one GlobalAveragePooling2D layer followed by four dense layers. Three of the four dense layers feature the ReLU activation function, while the last dense layer utilizes the softmax activation function with four output classes. The total number of trainable parameters for this model amounted to 3,674,628.

Like the CNN model, this model is also compiled with the Adam optimizer with the categorical cross-entropy loss. Training is also executed with a batch size of 32, a learning rate of 0.001, and is done for a total of 10 epochs.

3) VGG16Net

Visual Geometry Group-16 Network (VGG16Net) is a CNN architecture that originally has 16 layers of depth [18]. In this experiment, VGG16Net was employed via transfer learning, which involved freezing its ImageNet pre-trained base layers. This pre-trained base was then combined with a custom classification head, identical to the one utilized for the ResNet-50 model. The total number of trainable parameters for this VGG16Net configuration amounted to 2,101,764. A detailed visualization of the architecture is shown in Fig. 5.

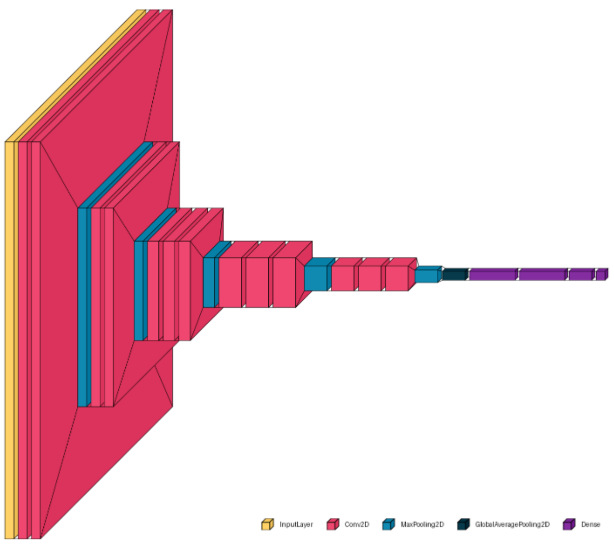

Fig. 5. Proposed VGG16Net architecture

IV. Results and Discussion

All classical machine learning (ML) models and deep learning (DL) models were trained and validated as described in the methodology. The experimental implementation was conducted in Python, leveraging libraries such as Scikit-learn [20] for ML models and TensorFlow [21] for DL models. Code execution was performed using the Google Colaboratory platform [22]. The hosted runtime utilized Google's Tensor Processing Unit (TPU) v2-8 accelerator [23], which was provided free of charge with limited usage, for all computations. The performance of each model was then compared and analyzed.

The DL models, except ResNet, showed a higher accuracy compared to the ML models. Specifically, CNN achieved the highest accuracy among all deep learning models, while Random Forest exhibited the highest accuracy among the classical machine learning models, as detailed in Table II.

TABLE II. Performance of trained ML and DL models

Model	Accuracy	F1 Score	Precision	Recall
k-NN	75.67	74.74	74.90	74.71
Random Forest	80.88	79.99	80.14	80.08
SVM	74.63	73.20	73.30	73.21
CNN	98.87	98.87	98.87	98.83
ResNet-50	55.83	49.40	73.76	34.79
VGG16Net	97.58	97.58	97.99	97.42

Analysis of the results reveals that the CNN model achieved an error rate of less than 2%, with VGG16Net demonstrating a comparable error rate of less than 3%. ResNet-50, in spite of being a DL model, could not effectively classify the diseases. ResNet-50 significantly underperformed, having more than 40% error rate, achieving accuracy lower than any of the ML models tested. Overall, the ML models performed less effectively than CNN and VGG16Net. Among the ML models, Random Forest achieved an error rate just under 20%, whereas *k*-NN and SVM both exhibited error rates exceeding 20%. The validation accuracy results are further visualized through confusion matrices presented in Figs. 6–11.

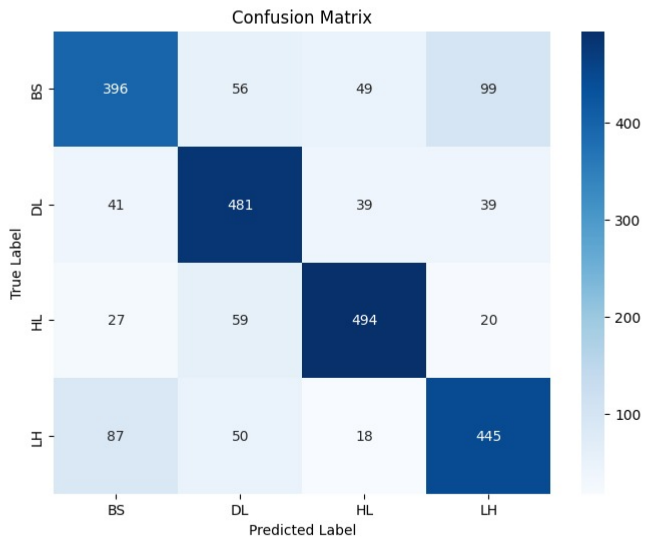

Fig. 6. The validation accuracy for the *k*-NN model in a confusion matrix

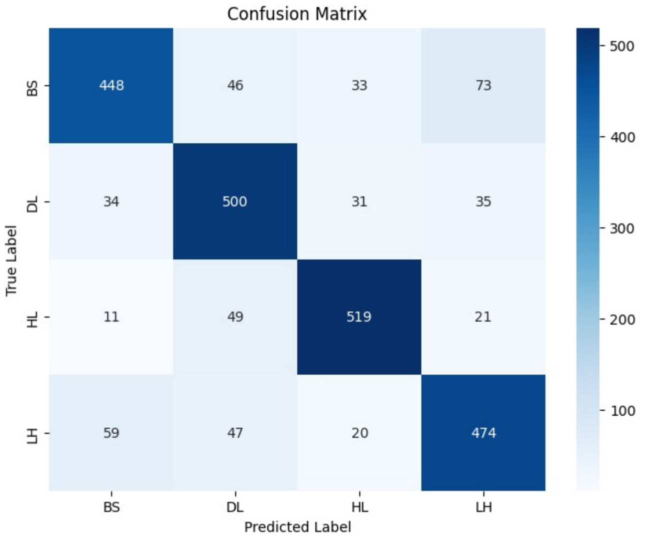

Fig. 7. The validation accuracy for the Random Forest model in a confusion matrix

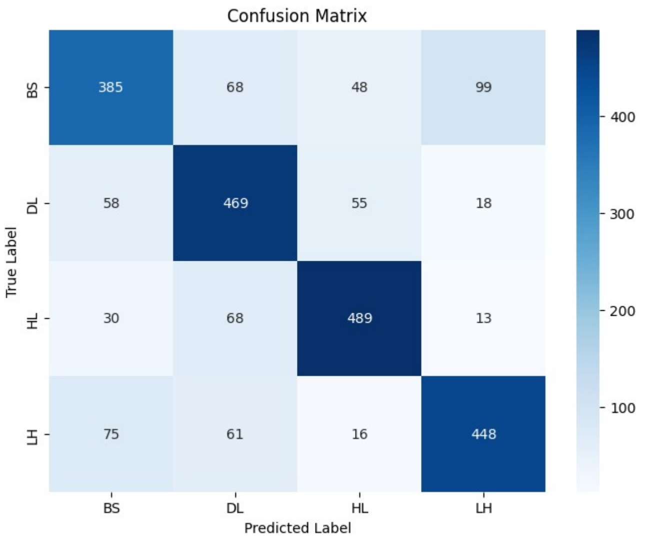

Fig. 8. The validation accuracy for the SVM model in a confusion matrix

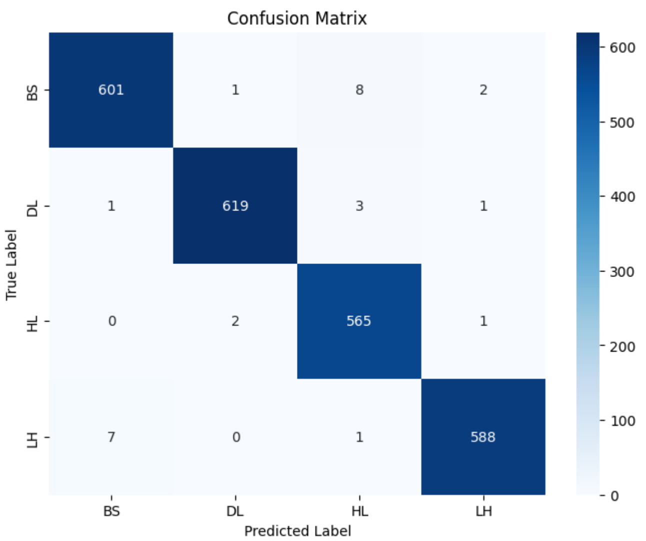

Fig. 9. The validation accuracy for the CNN model in a confusion matrix

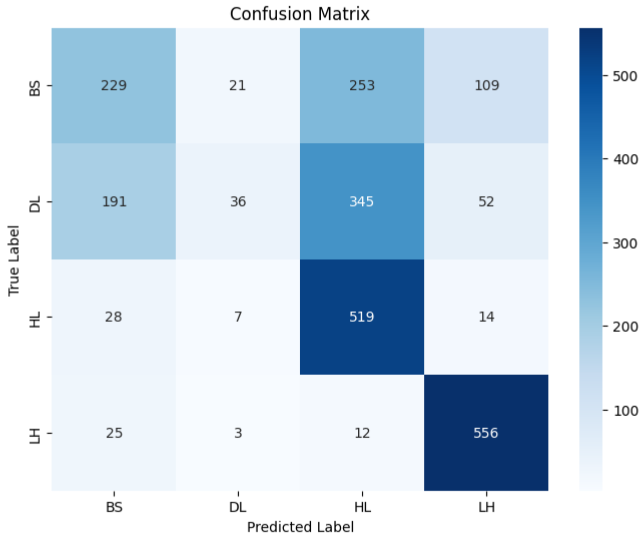

Fig. 10. The validation accuracy for the ResNet-50 model in a confusion matrix

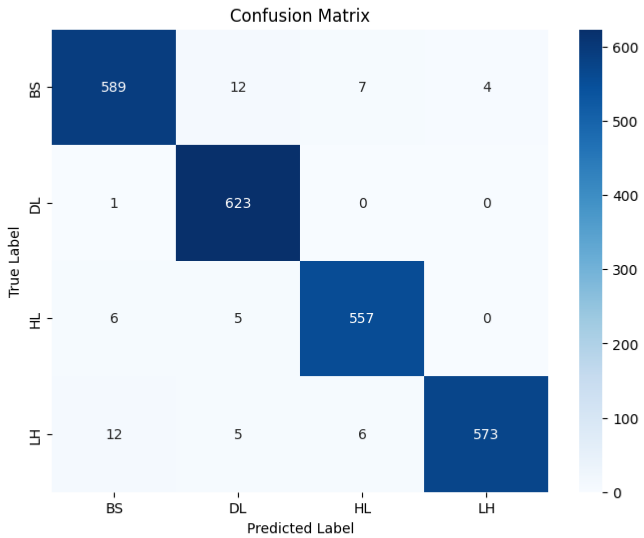

Fig. 11. The validation accuracy for the VGG16Net model in a confusion matrix

V. CONCLUSION

This research compares the performance of three ML methods and three DL methods in classifying rose leaf diseases. The study utilized *k*-NN, Random Forest, and SVM as ML models, while the DL models included a CNN and pre-trained architectures, specifically ResNet-50 and VGG16Net. Four classes were used for the classification: healthy leaves, black spot, leaf holes, and dry leaves. Initially, the dataset contained 3,113 images, but it was later augmented to a balanced total of 12,000 images to ensure equal representation across all classes. Before augmentation, the images underwent preprocessing steps including region segmentation, cropping, and resizing to 224×224 pixels. Normalization is applied before the images are input into the models. For the ML models, feature extraction was performed using GLCM, returning Contrast, Dissimilarity, Homogeneity, Energy, and Correlation. Data splitting was done with a 70:30 ratio into training and validation sets before they were finally used for training.

Results of the experiment suggest that two of the three DL methods, namely CNN and VGG16Net, outperformed the accuracy achieved by the classical ML models. However, the classical ML models employed in this experiment still performed consistently, even surpassing the accuracy of ResNet-50. The highest classification accuracy was achieved by CNN (98.87%), followed closely by VGG16Net (97.58%). Notably, Random Forest provided competitive results (80.88% accuracy) as a classical approach, particularly relevant in scenarios with limited computational resources or data availability.

Future research could focus on further improving the accuracy of classical approaches through experimentation with alternative models, including hybrid methods, and advanced image processing techniques. Additionally, the dataset could be expanded to include a greater variety of disease classes and more diverse image backgrounds, moving beyond the current isolated background. Finally, the integration of this solution into a practical, real-time disease detection application could be explored through an Internet of Things (IoT) implementation.

REFERENCES

[1] C. D. Badgujar, A. A. Bhagat, and V. S. Supe, “An economics analysis for export of fresh cut rose flowers from India,” *J Pharmacogn Phytochem*, vol. 2, pp. 291–298, 2019.

[2] M. Malik, W. Aslam, E. A. Nasr, Z. Aslam, and S. Kadry, “A Performance Comparison of Classification Algorithms for Rose Plants,” 2022, doi: 10.1155/2022/1842547.

[3] K. T. Venkatesha, A. Gupta, A. N. Rai, S. J. Jambhulkar, R. Bisht, and R. C. Padalia, “Recent developments, challenges, and opportunities in genetic improvement of essential oil-bearing rose (Rosa damascena): A review,” *Ind Crops Prod*, vol. 184, p. 114984, Sep. 2022, doi: 10.1016/J.INDCROP.2022.114984.

[4] “Top countries for Export of Roses.” Accessed: Apr. 21, 2025. [Online]. Available: https://www.nationmaster.com/nmx/ranking/export-of-roses

[5] R. U. Khan, K. Khan, W. Albattah, and A. M. Qamar, “Image-Based Detection of Plant Diseases: From Classical Machine Learning to Deep Learning Journey,” 2021, *Hindawi Limited*. doi: 10.1155/2021/5541859.

[6] S. Nuanmeesri, “A Hybrid Deep Learning and Optimized Machine Learning Approach for Rose Leaf Disease Classification,” 2021. [Online]. Available: www.etasr.com

[7] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, “Plant leaf disease detection using computer vision and machine learning algorithms,” *Global Transitions Proceedings*, vol. 3, no. 1, pp. 305–310, Jun. 2022, doi: 10.1016/j.gltp.2022.03.016.

[8] D. Radovanović and S. Đukanović, “Image-Based Plant Disease Detection: A Comparison of Deep Learning and Classical Machine Learning Algorithms,” *24th International Conference on Information Technology (IT)*, Feb. 2020.

[9] P. Wang, E. Fan, and P. Wang, “Comparative analysis of image classification algorithms based on traditional machine learning and deep learning,” *Pattern Recognit Lett*, vol. 141, pp. 61–67, Jan. 2021, doi: 10.1016/J.PATREC.2020.07.042.

[10] R. S. Chugh, V. Bhatia, K. Khanna, and V. Bhatia, “A comparative analysis of classifiers for image classification,” *Proceedings of the Confluence 2020 - 10th International Conference on Cloud Computing, Data Science and Engineering*, pp. 248–253, Jan. 2020, doi: 10.1109/CONFLUENCE47617.2020.9058042.

[11] Er. S. Kaur, “Image Classification Using Deep Learning,” *International Journal of Scientific Research in Engineering and Management*, vol. 07, no. 12, pp. 1–11, Dec. 2023, doi: 10.55041/IJSREM27484.

[12] H. Cheng, “Advancements in Image Classification: From Machine Learning to Deep Learning,” *ITM Web of Conferences*, vol. 70, no. 2016, 2025, doi: 10.1051/itmconf/20257002016.

[13] J. A. Pandian, V. Dhilip Kumar, O. Geman, M. Hnatiuc, M. Arif, and K. Kanchanadevi, “Plant Disease Detection Using Deep Convolutional Neural Network,” 2022, doi: 10.3390/app12146982.

[14] U. Barman, D. Sahu, and G. G. Barman, “A Deep Learning Based Android Application to Detect the Leaf Diseases of Maize,” in *Advances in Intelligent Systems and Computing*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 275–286. doi: 10.1007/978-981-15-8061-1_22.

[15] Jocelyn Dumlao, “Rose Leaf Diseases: Symptoms, ID & Control Guide,” Kaggle. Accessed: Apr. 21, 2025. [Online]. Available: https://www.kaggle.com/datasets/jocelyndumlao/rose-leaf-diseases-symptoms-id-and-control-guide/data

[16] Z. Jun and H. Jinglu, “Image segmentation based on 2D Otsu method with histogram analysis,” in *Proceedings - International Conference on Computer Science and Software Engineering, CSSE 2008*, 2008, pp. 105–108. doi: 10.1109/CSSE.2008.206.

[17] E. R. Reddy, S. D. Satturi, M. Harshini, and S. Shaik, “Rose Plant Leaf Disease Recognition Using Machine Learning Methodologies,” *Asian Journal of Research in Computer Science*, vol. 17, no. 11, pp. 65–72, Nov. 2024, doi: 10.9734/ajrcos/2024/v17i11519.

[18] K. Simonyan and A. Zisserman, “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION,” 2015. [Online]. Available: http://www.robots.ox.ac.uk/

[19] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016. Accessed: Jun. 21, 2025. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[20] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011, Accessed: Jun. 17, 2025. [Online]. Available: http://scikit-learn.sourceforge.net.

[21] T. Developers, “TensorFlow.” doi: 10.5281/ZENODO.15009305.

[22] D. S. R. Sukhdeve and S. S. Sukhdeve, “Google Colaboratory,” *Google Cloud Platform for Data Science*, pp. 11–34, 2023, doi: 10.1007/978-1-4842-9688-2_2.

[23] Y. Wang *et al.*, “Benchmarking the Performance and Energy Efficiency of AI Accelerators for AI Training,” in *Proceedings - 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing, CCGRID 2020*, Institute of Electrical and Electronics Engineers Inc., May 2020, pp. 744–751. doi: 10.1109/CCGrid49817.2020.00-15.