Project - Leaf Disease Classification

1. Abstract

This project aims on the analysis of CNN architectures by developing and evaluating deep learning models for classifying and detecting leaf diseases. We curated diverse and complex datasets from Mendeley, PlantVillage, and Rice Leaf, reducing image sets to manageable sizes and employing data preparation methods like cropping, resizing, and color jittering to boost model resilience and generalization. We trained and assessed three CNN models—ResNet18, VGG19, and MobileNetV2—using data augmentation and dynamic learning rate adjustments to minimize overfitting and improve performance. The models were evaluated on different datasets to analyze the impact of dataset complexity on performance, using accuracy, precision, recall, F1 score, t-SNE visualizations, and confusion matrices. Preliminary results indicate that while ResNet18 and MobileNetV2 consistently performed well, VGG19 struggled with more complex datasets. We performed fine-tuning hyperparameters, explored additional optimization methods, with the learning rate and analyzed the accuracy of the it with the three CNN models. We also applied transfer learning on two models.

2. Introduction and Problem Statement

Diagnosing leaf diseases is not just an essential part of crop health maintenance, but it is also a major component of agricultural economics and food supply chain management. By offering scalable solutions that might be integrated into real-time monitoring systems, advanced deep learning techniques have the potential to completely transform this field and significantly lower crop losses while increasing production. This research work mainly focuses on evaluating deep learning algorithms in detail to understand their effectiveness in classifying leaf diseases in different contexts.

There are many challenges in the way of developing a useful system for classifying leaf diseases. This is because disease patterns are diverse and complex, and different plant species have different symptoms. The system needs to be built to handle a wide range of data quality and adjust to newly discovered disease cases. The analysis of images is further complicated by environmental factors such as lighting and leaf condition. Because of the great analytical ability, deep learning models—most notably CNNs—are widely used; yet, they need large amounts of computational power as well as well-annotated datasets. Furthermore, they run the danger of overfitting and could perform poorly in the face of novel disease presentations, highlighting the necessity for models that can generalize well in a variety of scenarios.

To tackle these challenges, our methodology has included diverse data augmentation techniques to strengthen the models' resilience and generalization. Color jittering is crucial for countering issues with lighting and color variance in images. By using this technique, our models can detect diseases in a variety of environmental conditions and become less susceptible to changes in lighting. We also apply dynamic learning rate adjustments and data augmentation to enhance training and prevent overfitting. To assess how successfully the models distinguish between various classes of disorders, class separability is measured using t-SNE visualizations.

In particular, we focus on convolutional neural networks (CNNs) such as ResNet18, VGG19, and MobileNetV2, and we use deep learning models in our research. These models were selected due to their proven ability to identify images and their capacity to handle the high processing demands of large image datasets. Our approach involved meticulous data preparation where datasets from PlantVillage, Mendeley, and Rice Leaf were processed to represent a broad spectrum of disease conditions and leaf types. The training data was made more diverse and more realistically simulated by cropping, scaling, and adding augmentations like flipping and rotating to the images. Post-training, models are evaluated with accuracy, precision, recall, and F1 scores. Dynamic learning rate scheduling and transfer learning expedite the learning phase, while hyperparameter tuning optimizes CNNs for leaf disease detection.

Initial results show ResNet18 and MobileNetV2 performing well across datasets, with VGG19 being effective but less consistent. Utilizing transfer learning, we improved model accuracies, with notable enhancements seen in MobileNetV2 and VGG19 architectures, while hyperparameter tuning, particularly of learning rates and batch sizes, was critical in optimizing our CNN models for leaf disease classification. These findings indicate our comprehensive approach's versatility in capturing leaf disease diversity, setting a foundation for future advancements.

2.1. Related Work

Even though numerous deep-learning algorithms were invented for classification, none of them could surpass the capacity of convolutional neural networks (CNNs) in processing image data and extracting the relevant features making them the most used tool by resarchers for disease identification.

One of the major work in the field of plant disease classification using deep learning is by Mohanty et al. (2016)

[1] using Plant Village dataset in which he demonstrated high accuracy of 99.35% with deep convolutional neural network and identified 26 varieties of diseases in 14 crop species. The potential of deep learning technologies that have the power to revolutionize the agriculture sector enabling timely disease management by providing accurate and fast disease identification were revealed through the success of this project.

Ferentinos (2018) [15] achieved accuracy rates of more than 99% in the diagnoses of plant diseases through the application of various deep learning models. This study shows how important it is to have access to large and diverse datasets to properly train deep learning models. The study showed that with right amount of training data, CNNs could recognize plant diseases from images with high precision, which is necessary for developing scalable and robust agricultural disease detection systems.

Researchers investigated various CNN architectures to increase accuracy and efficiency. Various pre-trained models (AlexNet, VGG16, and ResNet50) were examined by Picon et al. (2019) [16] for identifying plant diseases. Although all models performed well, ResNet50 offers better trade off between accuracy and efficiency.

In order to locate diseased areas on plant leaves, Too et al. (2019) [17] incorporated region-based CNNs (R-CNNs) in his studies. This allows for more precise agricultural interventions. By providing more detailed diagnostic information, practical utility of deep learning models can be enhanced.

Combining deep learning with image processing proved to be successful, as shown by Kamilaris and Prenafeta-Bold u (2018) [18]. By combining conventional image processing techniques with CNNs, particularly in settings with complicated backgrounds and varied illumination, helped in improving feature extraction and classification accuracy. Combining these methodologies could enhance robustness and reliability.

Also, Barbedo's (2019) [19] investigation into the use of transfer learning has demonstrated potential in addressing the weakness of smaller datasets.Research achieved high accuracy on using pre-trained models on larger datasets and fine-tuning them for specific tasks. This approach highlighted the adaptability and scalability of deep learning models, which makes them more accessible for various agricultural applications.

All of these studies contribute to the field of plant detection using deep learning in a different way from each other, emphasising a collective result of what and all can be achieved with the usage of artificial intelligence in the agriculture field. They also highlight ongoing challenges and innovative solutions but also showcases the effectiveness of CNN's in various configurations, which further drives the research and application of these technologies in agricultural settings.

3. Proposed Methodologies

3.1. Dataset

The PlantVillage [1] contains approximately 54,000 leaf images spanning 38 classes. These images are available in various formats, including color, grayscale, and segmented forms, each with dimensions of 256x256 pixels and stored in '.jpg' format. To optimize computational efficiency, we streamlined the dataset to a size of 25414 images distributed across 15 classes. The dataset is divided into training, testing, and validation sets following an 80-10-10 ratio. In particular, there are 20,325 photos in the training set and 2,554 and 2,535 images in the test and validation sets, respectively.

Mendeley [2] dataset contains around 24,800 leaf images across 22 classes, this dataset offers high-resolution images exceeding 400x400 pixels. To streamline the training process and enhance manageability, the number of classes is reduced to 12, with 14894 images, maintaining a rich diversity in the dataset. The dataset is divided into training, testing, and validation sets following an 80-10-10 ratio. The training set encompasses 11,909 images, test and validation sets consist of 1,499 and 1,484 images, respectively.

Rice Leaf [3] dataset contains 5932 images spread across 4 unique classes. Each image within this dataset possesses dimensions of 300x300 pixels and is stored in '.jpg' format. The dataset is divided into training, testing, and validation sets following an 80-10-10 ratio. The training set comprises 4,745 images, the test and validation sets contain 595 and 592 images, respectively.

To enhance the integrity and uniformity of our input data, a comprehensive preprocessing strategy was implemented before training our deep neural network models. The images were uniformly resized to a 224x224 pixel resolution through random resized cropping for consistent processing across all samples, thereby allowing the neural network to converge more efficiently during training. Additionally, pixel intensity normalization was applied with a mean of ([0.485, 0.456, 0.406]) and a standard deviation of ([0.229, 0.224, 0.225]) to aid in stable model training. Data augmentation played a pivotal role in boosting the model's robustness against overfitting and included random horizontal and vertical flips, color jittering (adjusting brightness, contrast, saturation, and hue), random affine transformations (rotations, translations, and scaling), random perspective changes, and sporadic conversions to grayscale. The augmentation techniques expand the effective size of the dataset without the need for additional labeled data. This expanded variety stimulates the model to learn more general features instead of memorizing specifics. This enables our models to tackle the intrinsic variability of real-world

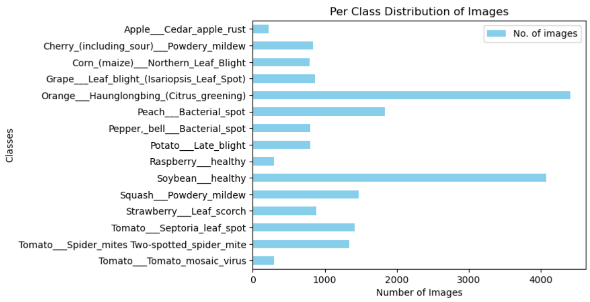

Figure 1. Class distribution for PlantVillage dataset

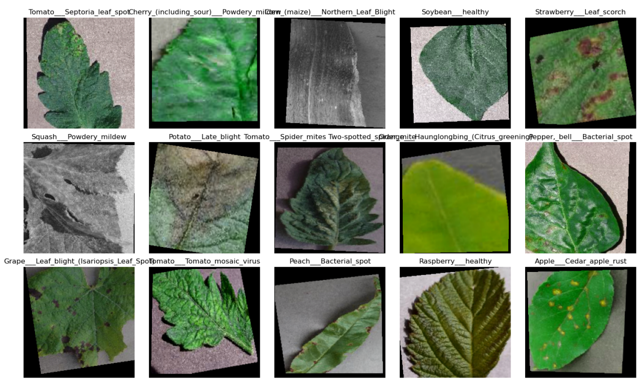

Figure 2. Example of PlantVillage dataset images

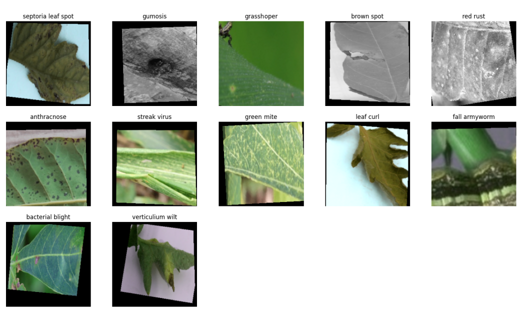

Figure 3. Example of Mendeley dataset images

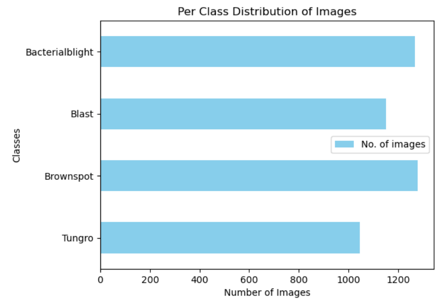

Figure 4. Class distribution for Rice leaf dataset

Figure 5. Example of Rice lea dataset images

data, like changes in lighting conditions or object orientations, by simulating these variations during training.

3.2. CNN Models

We have selected three CNN architectures for the leaf disease classification task: ResNet18, VGG19, MobileNetV2. ResNet18 stands for Residual Network. It consists of 18 layers including convolutional layers, pooling layers, fully connected layers, and shortcut connections. The model architecture comprises several residual blocks, each containing convolutional layers with batch normalization and ReLU activation functions. What differentiates ResNet from other CNNs is that it uses skip connections to pass the input over one or more layers. These connections help mitigate the vanishing gradient problem by allowing direct gradients flow. This is particularly advantageous for our task as we're dealing with a diverse dataset containing a variety of plant diseases, where deeper networks can capture more complex patterns and features.

VGG19 is composed of 19 layers (16 convolutional layers, 3 fully connected layers). It uses repeated blocks of convolutional layers with small receptive fields (3x3 with stride 1) followed by max-pooling layers. All hidden layers use the same padding to preserve spatial dimensions and ReLU activation functions. The network gets deeper through an increase in the number of filters after each max pooling. The final section includes two fully connected layers with 4096 channels each, and a final soft-max classifier. Even though it's computationally intensive compared to more modern architectures, its structure has demonstrated strong performance across various image classification tasks. This reliability makes it a suitable candidate for our task, where accuracy and stability are paramount.

MobileNetV2 is designed for mobile devices with constraints on computing power and memory. It improves upon the original MobileNet architecture with inverted residuals and linear bottlenecks. MobileNetV2 features an initial fully convolutional layer with 32 filters, followed by 19 residual bottleneck layers. Each bottleneck layer uses a depthwise separable convolution, splitting into a depthwise convolution and pointwise convolution to reduce computation. Unlike traditional residual mappings which go from low to high and back to low dimensions, MobileNetV2 inverts this by expanding the representation to a higher dimension, applies depthwise convolutions, and then projects back down to a low dimension. It ends with a 1x1 convolution, followed by average pooling and a fully connected layer with softmax for classification. Its inverted residual blocks and bottleneck layers ensure low latency and reduced memory usage without a substantial decrease in performance.

ResNet18 and VGG19 use greater resources than MobileNetV2 in terms of computational complexity. On the task of classifying leaf diseases, all three models perform quite well, with test accuracies estimated to be above 99%. ResNet18 and VGG19, on the other hand, are better suited for applications where high precision and processing power are not constraints, whereas MobileNetV2 is best suited for situations with limited resources, such as embedded and mobile devices.

Model	Training Time per epoch (s)	GFLOPS	Parameters (M)
ResNet18	1510.33	1.83	11.69
VGG19	1535.90	0.8551	143.67
MobileNetV2	1135.96	0.0145	3.5

Table 1. Computational complexities of models

3.3. Optimization Algorithms

Several crucial actions were done in order to validate and improve the CNN model. To ensure that the model was trained on a variety of samples and had distinct data for assessment, the dataset was first divided into training, validation, and test sets using a ratio of 80:10:10. This separation made it possible to evaluate the model's performance robustly and helped prevent overfitting.

SGD, or Stochastic Gradient Descent, is a widely-used optimization algorithm for training machine learning models, including deep neural networks. SGD introduces randomness into the optimization process by iteratively updating the model parameters using gradients derived from random mini-batches of data. Stability and convergence depend on the learning rate hyperparameter being carefully adjusted. Furthermore, to improve performance and avoid overfitting, SGD variations like SGD with momentum use regularization techniques and adaptively modify the learning rate.

We prioritized learning rate, weight decay, and momentum while choosing hyperparameters for our CNN model training using the SGD optimizer. The size of the step taken during parameter updates is determined by the learning rate, which also has a major impact on the training process's stability and pace of convergence. Higher learning rates can cause oscillations or divergence, whereas lower learning rates allow for more accurate updates but may cause delayed convergence. Weight decay, or L2 regularization, adds a penalty term to the loss function dependent on the size of the model's weights to assist prevent overfitting. Furthermore, momentum improves SGD by using data from earlier parameter changes to hasten convergence and stabilize training. To further fine-tune the model, a learning rate scheduler was also used to dynamically change the learning rate during training, lowering it when the validation loss plateaued.

In addition to training and validation accuracy, we also used training and validation loss to assess the optimization method. These metrics provide information on how well the model is learning and how broadly it can apply. We were able to measure convergence and identify possible overfitting or underfitting problems by keeping an eye on them over several epochs.

4. Results

4.1. Experiment setup

Our experiment aimed to meticulously develop, optimize, and evaluate three convolutional neural network (CNN) architectures: ResNet18, MobileNetV2, and VGG19, tailored for the precise task of leaf disease classification. The experimental process was methodically structured to ensure a comprehensive assessment of each model's capability in accurately identifying various leaf diseases.

Epoch	Train-Validation-Test	Batch	Criterion	Optimizer
10	80-10-10	32	Cross Entropy	SGD, Reduce Learning rate on plateu, Learning Scheduler

Table 2. Model parameters

The training of the models was conducted over 10 epochs, providing ample opportunity for the networks to learn and adapt to the intricate patterns present in the leaf images. We adhered to a standard data splitting ratio of 80/10/10 for training, testing, and validation sets, respectively, to ensure a balanced exposure to the data and to maintain an unbiased evaluation framework. The cross-entropy loss function was employed across all models to effectively measure the discrepancy between the predicted and actual disease classifications.

In terms of optimization, a learning rate of 0.001 was set as the starting point, with batch size maintained at 32 to strike a balance between computational efficiency and the accuracy of gradient estimation. Because SGD is resilient and simple, and because it worked well with our modestly small dataset and model complexity, we chose it over Adam. The training process was improved overall by our selection of hyperparameters, which included momentum for quicker convergence, a suitable learning rate to guarantee stability, weight decay for regularisation, and ReduceLROnPlateau to adaptively modify the learning rate. By making these choices, we were able to stabilise training, avoid overfitting, and eventually increase our models' capacity to generalise to new data. We have used learning schedular because without it we were getting fluctuating losses.

Validation of the models was executed through meticulous monitoring of training and validation loss and accuracy metrics, allowing for continuous assessment of the learning curve and model generalization. Precision, recall, and F1-measure were additional metrics utilized to provide a nuanced view of each model's performance, highlighting specific strengths and areas needing improvement in detecting and classifying leaf diseases.

Due to the computational demands of training deep CNNs we have used Colab with Kaggle to utilize its GPU capabilities to train the CNN architectures effectively for the task of leaf disease classification.

4.2. Main Results

In our study on leaf disease classification, we assessed the performance of three convolutional neural network models: ResNet18, MobileNetV2, and VGG19. These models were evaluated across three distinct datasets (D1:Plant Village, D2:Mendeley, D3: Rice Leaf) using metrics such as accuracy, precision, recall, and F1 score, to gauge their effectiveness in identifying various leaf diseases accurately.

Metric	ResNet18			MobileNetV2			VGG19		
	D1	D2	D3	D1	D2	D3	D1	D2	D3
Accuracy (%)	99	88	100	97	87	100	99	84	100
Precision (%)	97	86	100	97	80	100	100	86	100
Recall (%)	97	80	100	97	90	100	100	85	100
F1 score	97	81	100	97	89	100	100	85	100

Table 3. Evaluation of models

ResNet18 and MobileNetV2 show strong, consistent performance across all datasets, with ResNet18 edging out on D1 and D3, suggesting superior robustness. VGG19, however, varies, excelling on D3 but underperforming on D1 and D2, indicating it may require tailored adjustments for different data complexities. While D1 is well-handled by ResNet18 and MobileNetV2, D2 presents challenges that ResNet18 struggles with but are somewhat favorable for VGG19. D3, with perfect scores from all models, is likely the least complex. ResNet18 demonstrates excellent generalizability, and while MobileNetV2 is also effective, its performance shows variability. VGG19's best results on D3 reveal its potential when conditions are ideal. This analysis stresses the significance of matching datasets with model capabilities for optimal performance.

During the transfer learning we have choose specific combination of architecture with the dataset which has comparatively low accuracy and after performing transfer learning we found improvement on those combination. As shown in the below table, the accuracy of the MobileNet architecture for Plant Village dataset increased from 97 to 99%, While for VGG19 architecture (Mendeley dataset) increased from 84 to 89%.

Model		Train		Validation		Test
Dataset	Architecture	Loss	Accuracy	Loss	Accuracy	Accuracy
Plant Village	MobilenetV2	0.0691	97.76%	0.0075	99.72%	99.84%
Mendeley	VGG19	0.5212	80.72%	0.2959	89.00%	89.39%

Table 4. Transfer Learning

We have created the confusion matrix for all the dataset to generate the classification report and to see the classifications for the three architecture as shown in Figure 6 to understand the true positives, true negatives etc.

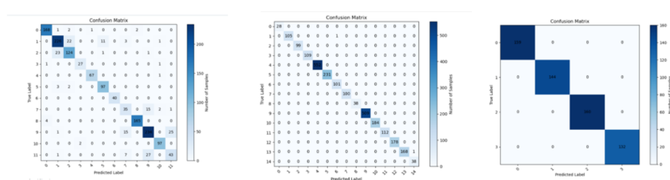

Figure 6. Confusion matrix for different architecture and datasets

From the confusion matrices provided for Mendeley, Plant Village and Rice Leaf datasets respectively, it appears that Rice Leaf dataset have clearer, more distinct patterns of correct classifications, as indicated by darker squares along the diagonal, which represent true positives. The rightmost matrix likely provides the best performance in terms of both accuracy and class separability for the Rice Leaf dataset.

The t-SNE visualization depicted shows the clusters of colors indicate the distinct groupings of the classes as shown in Figure 7. Almost all clusters appear well separated, suggesting that the RestNet18 architecture is able to differentiate perfectly between these classes effectively. While there is a degree of class separability, the overlap suggests that there could be room for improving the model's ability to differentiate between classes that are closely clustered.

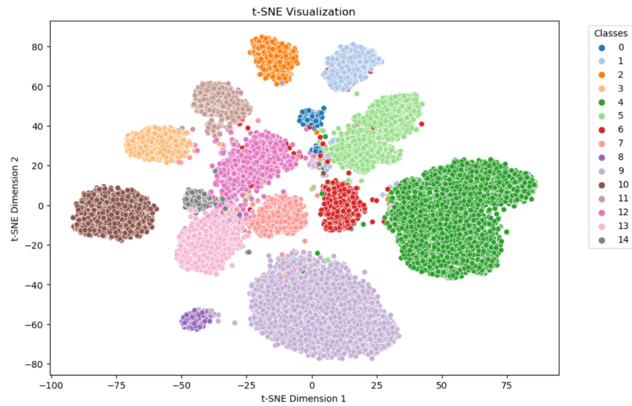

Figure 7. t-SNE graph for ResNet18 for Plant Village dataset

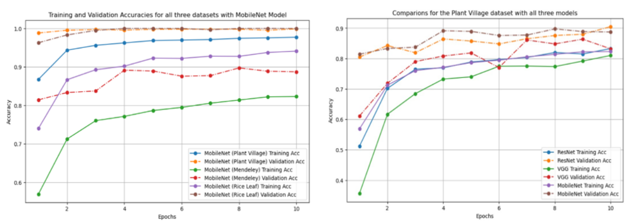

Figure 8. Model comparisons

In Figure 8, the first graph shows the training accuracy improving with more epochs for the MobileNetV2 Architecture, which is typical as the model learns from the data. Moreover, it shows that MobileNetV2 fits best for the Plant Village dataset and the least for Mendeley dataset when compared to their respective accuracy. The second graph on the right compares the accuracy of three models on Plant Village dataset, where we found out that the accuracy of ResNet18 Architecture for show the highest as compared to other. While the VGG19 Architecture showed the least accuracy as compared to the other two architectures.

4.3. Ablative study

Our ablative study focused on examining how different hyperparameters, specifically the learning rate and batch size, affect the performance of our CNN models in leaf disease classification. The learning rate and batch size are crucial factors that influence the training process and the model's ability to generalize from the training data.

Our tests show a significant pattern in the VGG19's performance with respect to changes in learning rates as shown in Figure 9. We find that accuracy decreases when the learning rate is increased incrementally from 0.001 to 0.004. The observed decrease implies that increased learning rates cause overshooting of the ideal parameter values during optimisation. When we have very high learning rate like 0.1 the accuracy was very suboptimal as shown.

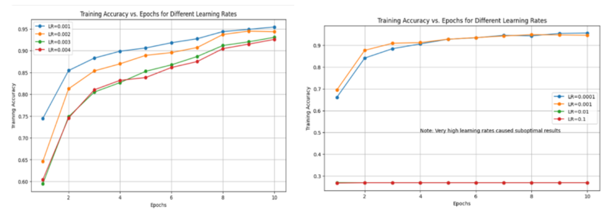

Figure 9. Hyper parameter tuning - Learning Rate

Architecture and Dataset	Learning Rate	Accuracy(%)
VGG19 - Mendeley	0.001	95.43
	0.002	94.43
	0.003	93.06
	0.004	92.59
ResNet18 - Plant Village	0.001	99.99
	0.002	98.51
	0.003	97.35
	0.004	97.56
MobileNetV2 - Rice Leaf	0.001	95.43
	0.002	94.34
	0.003	93.66
	0.004	92.59

Table 5. Hyper parameter optimization - Learning Rate

While analyzing the tuning of batch size from 32 to 64, in case of MobileNetV2 (Mendeley dataset) the accuracy get increased. Although, in case of Resnet18 (Mendeley dataset), it decreases when the batch size is increased (88.26% to 87.26%). For VGG19 (Rice leaf dataset) the accuracy drop slighlty. From This we can infer that larger batch sizes might not always lead to improved performance.

Model and Dataset	Batch Size	Accuracy(%)
Resnet18 - Mendeley	32	88.26
	64	87.26
MobilenetV2 - Mendeley	32	87.44
	64	89.11
VGG19 - Rice Leaf	32	100
	64	99.81

Table 6. Hyper parameter optimization - Batch Size

The optimal learning rate and batch size were found to vary depending on the specific model and dataset. For instance, ResNet18 and MobileNetV2 showed improved accuracy with a moderate batch size and a lower learning rate, highlighting the need for a balanced approach to hyperparameter tuning. On the other hand, VGG19's performance was highly sensitive to the learning rate adjustments, with an increase in accuracy at a lower learning rate.

5. References

1. SP Mohanty. PlantVillage Dataset. Available at: https://github.com/spMohanty/PlantVillage-Dataset.

2. Patrick Mensah Kwabena, Vivian Akoto-Adjepong, Kwabena Adu, Mighty Abra, Ayidzoe Elvis Asare Bediako, Owusu Nyarko-Boateng, Samuel Boateng, Esther Fobi Donkor, Faiza Umar Bawah, Nicodemus Songose Awarayi, Peter Nimbe, Isaac Kofi Nti, Muntala Abdulai, Remember Roger Adjei, Michael Opoku. Dataset for Crop Pest and Disease Detection. DOI: 10.17632/bwh3zbpkpv.1. Available at: https://data.mendeley.com/datasets/bwh3zbpkpv/1.

3. Prabira Kumar Sethy. Rice Leaf Disease Image Samples. DOI: 10.17632/fwcj7stb8r.1. Available at: https://data.mendeley.com/datasets/fwcj7stb8r/1.

4. P. Sirenjeevi, J. Madhu, Karthick K., Agalya R., Srikanth T., Elangovan T., and Nareshkumar R. Leaf Disease Identification using ResNet. DOI: 10.1109/ICECONF57129.2023.10083963. Available at: https://ieeexplore.ieee.org/document/10083963.

5. Elhoucine Elfatimi, Recep Eryigit, Lahcen Elfatimi. Beans Leaf Diseases Classification Using MobileNet Models. DOI: 10.1109/ACCESS.2022.3142817. Available at: https://ieeexplore.ieee.org/document/9681085.

6. Dr. Thanh-Hai Nguyen, Dr. Nghia Thanh Nguyen, Mr. Ba-Viet Ngo. A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease. Available at: https://www.mdpi.com/2624-7402/4/4/56.

7. Jinzhu Lu, Lijuan Tan, Huanyu Jiang. Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification. https://doi.org/10.3390/agriculture11080707.

8. Wubetu Barud Demilie. Plant disease detection and classification techniques: a comparative study of the performances. Available at: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00863-9.

9. Plant Disease Detection using AI-based VGG-16 Model. Available at: https://thesai.org/Downloads/Volume13No4/Paper_84-Plant_Disease_Detection_using_AI_based_VGG_16_Model.pdf.

10. Prechelt, L. (1998). Early Stopping - But When?. In Neural Networks: Tricks of the Trade. Springer, Berlin, Heidelberg. https://link.springer.com/chapter/10.1007/3-540-49430-8_3.

11. Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427-437. https://www.sciencedirect.com/science/article/abs/pii/S0306457308001608.

12. Shah SR, Qadri S, Bibi H, Shah SMW, Sharif MI, Marinello F. Comparing Inception V3, VGG 16, VGG 19, CNN, and ResNet 50: A Case Study on Early Detection of a Rice Disease. Agronomy. 2023; 13(6):1633. https://doi.org/10.3390/agronomy13061633

13. Ferentinos, K. (2018). "Deep learning models for plant disease detection and diagnosis." *Computers and Electronics in Agriculture*, 145, 311-318. DOI: https://doi.org/10.1016/j.compag.2018.01.009.

14. Picon, A., Alvarez-Gila, A., Seitz, M., Ortiz-Barredo, A., Echazarra, J., & Johannes, A. (2019). "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild." *Computers and Electronics in Agriculture*, 161, 280-290. DOI: https://doi.org/10.1016/j.compag.2019.05.004.

15. Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). "Automated plant diseases diagnosis using image segmentation and SVM techniques." *Computational Intelligence and Neuroscience*, 2019. DOI: https://doi.org/10.1155/2019/6731861.

16. Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). "Deep learning in agriculture: A survey." *Computers and Electronics in Agriculture*, 147, 70-90. DOI: https://doi.org/10.1016/j.compag.2018.02.016.

17. Barbedo, J. G. A. (2019). "Factors influencing the use of deep learning for plant disease recognition." *Biosystems Engineering*, 172, 84-91. DOI: https://doi.org/10.1016/j.biosystemseng.2018.05.013.

6. Appendix

D1:	PlantVillage Dataset
D2:	Mendeley Rice Leaf Dataset
D3:	Rice Leaf Diseases Dataset
CNN:	Convolutional Neural Networks
SGD:	Stochastic Gradient Descent