Project - Leaf Disease Classification

1. Introduction and Problem Statement

Advancements in computer vision and deep learning can revolutionize the agricultural industry by automating disease detection, aiming to develop a reliable, efficient system for global food systems, addressing challenges like leaf diseases.

We had to optimize the dataset for both size and quality, make sure the model was resilient to variations in the data, mitigate overfitting by adjusting sample sizes, and dynamically modify the learning rate to speed up convergence and avoid bad solutions, among other challenges, in order to improve the model's performance. Strategies including data augmentation, incremental sample size adjustments, and dynamic learning rate alterations were used to overcome these challenges.

The aim is to create a model that achieves high accuracy and robust generalization for leaf disease classification. We plan to use common assessment criteria, such as accuracy, precision, recall, and F1 score, to assess our findings. Furthermore, we want to evaluate class separability using TSNE visualizations and class-wise performance using confusion matrices.

2. Proposed Methodologies

2.1. Dataset

The PlantVillage[1] dataset contains approximately 54,000 leaf images across 38 classes. These images are available in color, grayscale, and segmented forms, each with dimensions of 256x256 pixels and stored in '.jpg' format. To manage computational resources efficiently, we have reduced the dataset size to 11,400 images, containing 300 images per class. This dataset serves as a fundamental resource for training and validating our models due to its extensive coverage of plant diseases.

The Mendeley[2] dataset consists of around 24,800 leaf images categorized into 22 classes. Each image has dimensions exceeding 400x400 pixels and is stored in '.jpg' format. We have lowered the number of classes to 12 in order to expedite our training procedure and guarantee a more concentrated and manageable dataset. This dataset strengthens the variety of our training set and increases the resilience of our models.

The Rice Leaf[3] dataset comprises 5,900 images distributed across 4 distinct classes. Each image within this dataset possesses dimensions of 300x300 pixels and is stored in '.jpg' format. The diversity of our training data is enhanced by the Rice Leaf dataset, which is lower in size than the other datasets but provides unique insights on diseases that impact rice plants.

2.2. Steps for Pre-processing

The preprocessing pipeline includes processes to enhance the training data quality and variety. Data augmentation techniques, such as resizing, cropping, flipping, and rotating, create more variable photos with varying visuals, improving generalization and reducing overfitting by exposing the model to a wider range of variables.

We utilized color jittering to regulate image saturation, contrast, and brightness, replicating real-world lighting conditions and color tones, thereby strengthening the model's resistance to external influences affecting image appearance.

We used affine transformations like translation, scaling, and shearing to replicate distortions from shifting perspectives or camera angles, enhancing the model's ability to learn invariant characteristics across different orientations.

The pipeline for preprocessing incorporates random grayscale conversion, creating variants resembling the dataset's grayscale photos. This strategy adds diversity to the input data distribution, enhancing model learning efficiency.

2.3. CNN Models

The ResNet18, or Residual Network with 18 layers, is renowned for its efficiency and simplicity. Deeper networks can be trained since it uses residual blocks to address the vanishing gradient issue. ResNet18 is a good choice for a variety of image classification tasks because of its balance between model complexity and performance, which is achieved by its modest depth, computational efficiency, and around 11.7 million parameters.

VGG19, an architecture that is a variation of the VGG (Visual Geometry Group) family, is distinguished by its consistent design with 19 layers, including 3 fully connected layers, 16 convolutional layers, and 143.7 million parameters. Deep convolutional stacks, a characteristic of VGG networks, allow them to extract complex information from input pictures. When a lot of processing power is available, VGG19 performs exceptionally well in picture recognition tasks, even with its depth and parameter-heavy structure.

MobileNetV2, with its lightweight and efficient architecture, is well-suited for embedded and mobile devices. It has depthwise separable convolutions, which significantly lower the computational cost of the model without sacrificing its representational capability. It has about 3.4 million parameters. Because MobileNetV2 strikes a good balance between accuracy, latency, and model size, it is a good choice for real-time applications and situations with limited resources.

2.4. Evaluation of the Model

We performed a thorough evaluation of our CNN models using a rigorous procedure, evaluating measures including accuracy, precision, recall, and F1-score using a separate validation set. This methodology facilitated a thorough comprehension of the performance attributes of every model. Furthermore, impartial assessment was assured and insights into the models' generalization skills were obtained by testing on a different dataset. A additional help in pinpointing certain areas for improvement and optimization was provided by confusion matrix analysis. Additionally, we used t-SNE visualization to investigate feature distribution in a reduced-dimensional space, which allowed us to evaluate model representations qualitatively.

3. Attempts at solving the problem

As of now, we have implemented 9 models. We have trained and evaluated each three datasets with ResNET18, Vgg19 and MobileNETV2 architecture.

Using a cross-entropy loss function and a batch size of 32, we trained the model across ten epochs. Using a learning rate decrease on plateau, the stochastic gradient descent (SGD) optimizer was used for the training. A split of the data was made into test(10%), validation(10%), and training(80%) sets. Using a dropout layer (dropout rate of 0.7) in place of the fully connected layer, and then a linear layer to match the number of classes in the dataset.

ResNet18 achieved test accuracies of 96.62% and 100% on the PlantVillage and Rice Leaf datasets, respectively, displaying impressive performance. This achievement highlights how well ResNet18 captures complex patterns in the datasets, resulting in better disease classification.

On the other hand, VGG19 has trouble producing adequate results while using the PlantVillage dataset. This discrepancy shows that the PlantVillage dataset's complexity may outweigh the depth of the VGG19 architecture. On the other hand, MobileNetV2 performed admirably on each dataset. With its depth-wise separable convolutions and lightweight architecture, MobileNetV2's efficiency allows it to collect features even in contexts with limited resources.

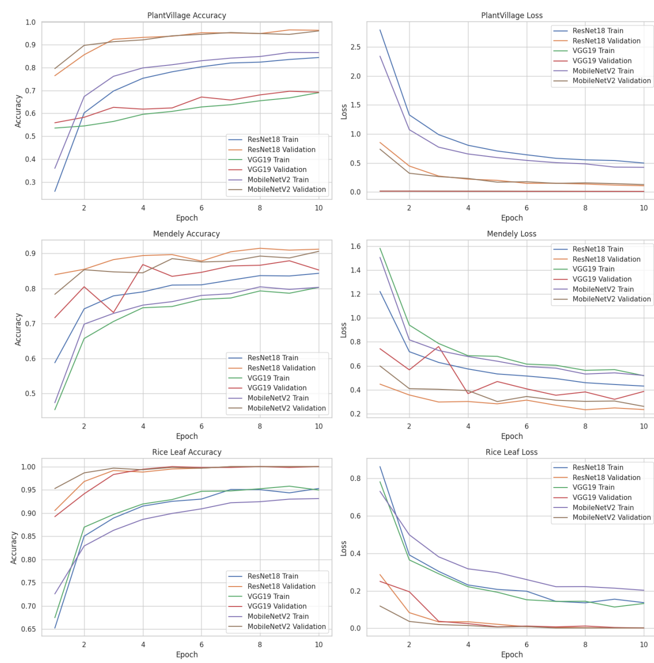
These results demonstrate that while lighter designs like MobileNetV2 are useful in situations when computing resources are limited, as seen in the CCMT dataset, deeper architectures like ResNet18 are superior at capturing complicated patterns in some datasets like Rice Leaf. The deeper design of VGG19 may still be useful for datasets with distinct properties.

Table 1. Model parameters

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

Table 2. Evaluation of models

Metric	ResNet18			MobileNetV2			VGG19		
	D1	D2	D3	D1	D2	D3	D1	D2	D3
Accuracy (%)	97	86	100	97	89	100	69	88	100
Precision (%)	97	86	100	97	80	100	71	86	100
Recall (%)	97	80	100	97	90	100	68	88	100
F1 score	97	81	100	97	89	100	67	87	100



4. Future Improvements

Adjusting hyperparameters like learning rate, dropout rate, and batch size might maximize model performance and increase the model's accuracy. We plan to try other optimization techniques, such as Adam or RMSprop, in addition to SGD. Using transfer learning to improve model accuracy is a promising approach. The model can profit from learnt features and adapt to the particular job at hand by using pre-trained models as feature extractors and fine-tuning them on the target dataset. Only a part of the pre-trained model's weights need to be updated during fine-tuning in order to preserve the important information from the original job.

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. Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. In Advances in neural information processing systems (pp. 3320-3328). https://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.

13. Namani, Sahithya & Akkapeddi, Lalita & Bantu, Saritha. (2022). Performance Analysis of VGG-19 Deep Learning Model for COVID-19 Detection. 10.23919/INDIACom54597.2022.9763177. https://www.researchgate.net/publication/360768508_Performance_Analysis_of_VGG-19_Deep_Learning_Model_for_COVID-19_Detection.

14. 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

6. Appendix

D1: PlantVillage Dataset
D2: Mendeley Rice Leaf Dataset
D3: Rice Leaf Diseases Dataset