Tomato Leaf Disease Classification

1. Introduction and problem statement

Tomatoes are one of the most commonly harvested crops, and they are consumed by people all over the world. However, leaf diseases are a serious threat to both the quality and quantity of tomato yields which directly reflects in the economic loss according to statistics [4]. Leaf disease identification is a resource-consuming process when done manually, and it also requires the help of an expert to classify the leaf diseases. Early and accurate diagnosis of these diseases is critical for maximizing yields. Our proposed methodology uses Convolutional Neural Networks (CNNs) to develop a deep learning-based approach for the accurate classification of tomato leaf diseases, aiming to revolutionize disease management practices.

One of the principal challenges in the dataset is the numerous types of noise in the dataset, which hinders the accuracy of the model. The images have different illumination levels and backgrounds within a class in a dataset, which makes it hard for the model to extract relevant features from images. Additionally in dataset 2, we found the presence of duplicate images that have been preprocessed to enhance the dataset's diversity. Hence the accurate preprocessing steps are crucial for achieving high accuracy.

The primary goal of the project is to investigate the impact of dataset sizes, optimize hyperparameters, and evaluate the effectiveness of transfer learning, contributing valuable insights to enhance the model's performance and applicability. We will employ a series of performance metrics, including accuracy, precision, recall, and the F1 score, to thoroughly assess the effectiveness of our models. These metrics will provide a comprehensive view of our models' performance, highlighting their strengths and identifying areas needing improvement. Additionally, the use of confusion matrices will allow us to visualize the models' classification accuracy across different disease types, offering insights into any patterns of misclassification. Finally, we aim to refine our models to achieve optimal performance. Our proposed model promises quick and accurate classification of tomato leaf diseases, eliminating the need for specialized agricultural expertise.

2. Proposed Methodologies

After carefully reviewing numerous datasets on platforms such as Kaggle and other reputable websites, we chose the three datasets listed above to be used for our project. Each dataset was chosen based on its unique characteristics and relevance to our project objectives. The first dataset contains resized images of Taiwan tomato leaves for consistency and ease of analysis. The second dataset, on the other hand, contains a wide range of images of tomato leaves collected in both laboratory and natural field conditions, increasing its variety. The third dataset is the largest of the three, containing a diverse set of crop images from which we extracted specific tomato leaf disease categories. Table 1. below summarizes the characteristics of all the datasets.

Dataset	Images	Dimensions	Classes
Dataset [1]	2k	227 x 227	3
Dataset [2]	14.6k	256 x 256	6
Dataset [3]	18.1k	256 x 256	10

Table 1. Dataset Description

We implemented a customized method for data preprocessing to raise the quality of preprocessed images and enhance the models' performance. When comparing the train data to the validate and test sets, we used a variety of different techniques, such as resizing, random horizontal flip, random rotation, ColorJitter, and normalizing with standard deviation and mean values. We applied resize and normalize techniques to the test and validate set. Utilizing various preprocessing techniques helps in optimizing the variations between the train and validate sets. The model will validate and test more accurately due to the variations in the images.

We trained three CNN architectures: ResNet18, VGG16, and MobileNetV2, across all the 3 datasets. This resulted in the creation of 9 distinct models, each tailored to maximize the unique strengths of its respective architectures. ResNet18 is a relatively newer architecture that performs well on image classification tasks due to its unique feature 'Residual Learning', which addresses the vanishing gradient problem using the shortcut connections [7]. VGG16 has dropout layers at the end which helps in preventing the overfitting of the model and thus giving good performance, it also helps in reducing the spatial dimension [5]. MobileNetV2 uses Depth-wise Separable Convolutions which use Inverted residuals to preserve the information and It can give a good accuracy with the low parameters and mathematical operation that can help using deep learning in mobile devices [6].

The following hyperparameters are incorporated on all our models, Batch size: 64, Loss: Cross-Entropy Loss, Optimizer: Adam. For the optimizer, we found 0.001 to be the optimal value for the learning rate as it's relatively less which helps the model converge smoothly without overshooting the minima and thus limits the risk of overfitting. For training, the number of epochs we are using is fixed to 15, and to reduce the risk of overfitting, we adopted an early stopping criterion which stops the training if the validation loss is the same or increasing for consecutive epochs until the threshold is reached. This significantly improved the model's overall performance.

3. Attempts at solving the problem

Using three datasets and ResNet18, MobileNetV2, and VGG16 for training, we have successfully trained nine models. There are three classes in Dataset 1, with MobileNetv2 having the highest accuracy at 81.9% and ResNet18 coming in second at 81.5%. But VGG16 lags with the lowest accuracy of 72.16%, especially when it comes to correctly classifying images in the late blight class. These images' varied backgrounds could have made it more difficult for the model to identify them accurately.

Upon comparing the results obtained from dataset 2, we found that mobilenetv2 performs the best with an accuracy of 83.6% followed by Resnet18 with an accuracy of 82.3%, and lastly VGG16 with an accuracy of 80.96%. The VGG16 model had some trouble classifying the 'late blight' as observed from the confusion matrix.

MobilNetv2 is the best performer in Dataset 3, exhibiting an accuracy of 91%. Conversely, VGG16 performs worse, having an accuracy of only 85.6%. This might be explained by the dataset's inclusion of more than 18,000 photos and 10 classes. The models are trained on a variety of images, which improves their accuracy. When it comes to correctly classifying the image of a yellow leaf curl, VGG16 has demonstrated impressive performance. Unlike some other models, though, it has had trouble accurately classifying mosaic virus and early blight. Compared to the other classes, the mosaic virus class has a notably smaller number of images. One explanation could be that the model finds it challenging to correctly classify the images in these classes due to noisy data and low image quality.

The below plots represent the loss and accuracy comparison of all the models trained using the chosen datasets.

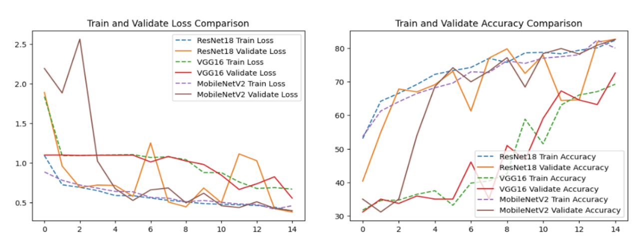

Figure 1. Dataset 1 Train and Loss Comparisons All Models

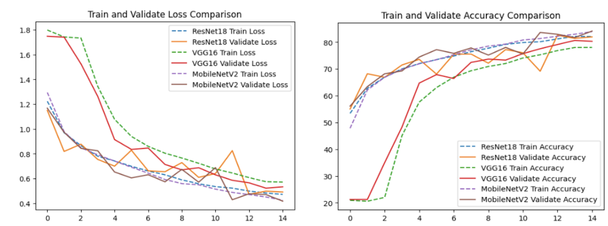

Figure 2. Dataset 2 Train and Loss Comparisons All Models

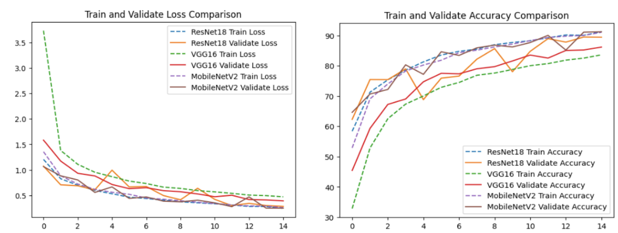

Figure 3. Dataset 3 Train and Loss Comparisons All Models

The below table represents the classification report containing the metrics for all the models trained using the chosen datasets.

Model	Dataset	Recall (%)	F1 Score (%)	Precision (%)	Accuracy (%)
ResNet18	1	81.5	81.4	81.3	81.5
MobileNetv2	1	81.9	81.6	81.7	81.9
VGG16	1	62.1	61.2	60.3	72.16
ResNet18	2	82.3	82.7	83	82.3
MobileNetv2	2	83.6	84.5	85.4	83.6
VGG16	2	85.6	85.7	85.9	80.96
ResNet18	3	89.5	90.1	90.6	89.5
MobileNetv2	3	91	91.3	91.6	91
VGG16	3	85.6	85.7	85.9	85.6

Figure 4. Metrics Analysis for All Models

4. Future Improvements

From the above trained 9 models, we would be addressing the performance issues and thereby choosing two least performing models and attempting to increase their performance through optimization. Additionally, we would be choosing two of the best performing models and applying transfer learning to them. Finally, all the metrics will be plotted using t-SNE plots.

References

[1] Tomato leaf disease dataset 1. https://www.kaggle.com/datasets/joseenriquelopez/tomato-leaf-diseases. 1

[2] Tomato leaf disease dataset 2. https://www.kaggle.com/datasets/cookiefinder/tomato-disease-multiple-sources. 1

[3] Tomato leaf disease dataset 3. https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset. 1

[4] H. D. Gadade and D. K. Kirange. Tomato leaf disease diagnosis and severity measurement. *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2020. 1

[5] M. Gehlot and M. L. Saini. Analysis of different cnn architectures for tomato leaf disease classification. *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, 2020. 1

[6] M. H. Anuz R. Moyazzoma, M. A. A. Hossain and A. Sattar. Transfer learning approach for plant leaf disease detection using cnn with pre-trained feature extraction method mobilnetv2. *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 2021. 1

[7] N. Sharma R. Singh and R. Gupta. Classification and detection of corn leaf disease using resnet 18 transfer learning model. *2023 8th International Conference on Communication and Electronics Systems (ICCES)*, 2023. 1

CVPR
#G

A. Supplementary Material

Figure 5. Sample Images in Dataset 1

Figure 6. Sample Images in Dataset 2

Figure 7. Sample Images in Dataset 3