Tomato Leaf Diseases Classification and Real-World Application

Radhey Ruparel (rr85), Judy Fang (cf52), Rung-De Chu (rc118)

Abstract

In the pursuit of advancing agricultural technology and enhancing the efficiency of plant disease detection systems, this study focuses on employing Convolutional Neural Networks (CNNs) and transfer learning techniques, specifically utilizing ResNet models, to accurately and swiftly identify diseases in tomato plant leaves. Recognizing the critical role of tomatoes in both nutritional and economic sectors globally.

Our experimental journey began with the application of a Multilayer Perceptron (MLP) model, which achieved an accuracy of 82.00%, aligning with the performance of other machine learning models cited in recent literature. Recognizing the necessity for more sophisticated models to meet our objectives, we shifted our focus to developing a CNN model. This model demonstrated a significant improvement, achieving an accuracy of 95.13%, thereby surpassing the benchmarks set by previous studies. Despite the success of the CNN model in terms of accuracy, its practical application in real-time scenarios was hindered by its computational complexity and latency, leading us to explore the potential of transfer learning with the ResNet18 architecture as a solution, providing an accuracy of 99.41%. The primary goal of this research is to develop an optimized deep-learning framework that combines the accuracy benefits of CNNs with the efficiency and real-world applicability provided by transfer learning techniques. The anticipated outcome of this study is a lightweight, highly accurate model with low latency capable of real-time disease classification, offering a significant advancement over existing approaches and a valuable resource for the agricultural sector.

1 Introduction

Detecting diseases in tomato plants is crucial in the United States, a leading producer and consumer of tomatoes. The health of tomato crops is important to the country's agricultural economy. The United States Department of Agriculture (USDA) indicates tomatoes as one of the most valuable grown vegetables. Tomato leaf disease significantly impacts crop yield and economic outcomes for farmers, necessitating rapid detection and management to sustain tomato production and enhance agricultural profitability. Traditional manual identification methods are labor-intensive and subject to variability due to subjective factors, including fatigue and mood. Consequently, computer vision-based recognition of tomato leaf diseases has emerged as a practical approach, benefiting from advancements in computer technology and image recognition capabilities, thereby offering a more reliable and efficient solution for disease management in agriculture.

1.1 Why does the Accurate Classification of Tomato Plant diseases matter?

Tomatoes are a highly nutritious crop with significant worldwide implications for the agricultural economy. Their cultivation and production levels are crucial, given tomatoes' nutritional and pharmacological benefits, which include the prevention of hypertension, hepatitis, and gingival bleeding, among others [1]–[6]. The widespread utilization of tomatoes has led to increasing demand. Notably, small farmers, who account for over 80% of agricultural production, face losses exceeding 50% due to pests and diseases [7]–[8]. Thus, identifying tomato diseases and pests is critical to ensuring crop health and yield [9].

However, traditional manual methods for detecting pests and diseases are inefficient and costly [10]. The advancement of the Internet and the adoption of image-based disease identification techniques in computer vision have revolutionized this area. By leveraging efficient image identification technologies, these approaches enhance recognition efficiency, reduce operational costs, and improve accuracy [11].

1.2 Broad Overview

The motivation for selecting plant disease detection as the core of our research is driven by the urgent requirement to enhance agricultural practices amid growing global food demands.This study integrates state-of-the-art deep learning and transfer learning techniques with the objective of surpassing traditional machine learning models in accuracy, reducing model training durations, and diminishing latency to facilitate real-time detection. Such advancements are envisioned to provide substantial benefits to farming communities, thereby contributing to the overarching goal of fostering sustainable agricultural practices. Through equipping farmers with efficient disease management tools, this initiative aims to bolster the resilience of food supply chains.

Addressing the exigent demand for precise plant disease identification, our research employs a spectrum of Deep Learning strategies, ranging from standard multi-layer neural network classifiers to sophisticated Convolutional Neural Networks (CNNs). CNNs demonstrate exceptional proficiency in discerning intricate patterns within plant imagery, a capability critical for the accurate diagnosis of diseases. To further our objectives of reducing latency and enhancing diagnostic accuracy, this study incorporates transfer learning methodologies into CNN frameworks, thereby yielding considerable benefits. By leveraging established models such as ResNet through transfer learning, empowers our system to leverage pre-trained networks, extracting valuable knowledge from diverse datasets and effectively applying it to our specific domain, thereby amplifying the diagnostic capabilities of our models.

2 Related Work

- In a recent investigation, a comprehensive comparative analysis was undertaken involving six distinct machine learning algorithms—namely, Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), k-Nearest Neighbors (k-NN), Naïve Bayes (NB), and Linear Discriminant Analysis (LDA)—to address the challenge of tomato leaf disease classification. These algorithms functioned as foundational classifiers, attaining classification accuracies of 89.81%, 79.81%, 91.53%, 85.91%, 44.42%, and 82.84%, respectively. Subsequently, the study explored the efficacy of ensemble methods, specifically the simple majority and weighted majority voting techniques, as a means to enhance classification performance. Applying these ensemble methods resulted in accuracies of 93.49% and 95.58%, respectively. The ensemble methods, through the aggregation of predictions from multiple base classifiers, exhibit superior classification performance compared to the utilization of individual models in isolation [12].

- One of the significant challenges in training deep learning models for such classification problems is the computational cost, especially when using CPUs. Training on CPUs can be significantly slower, particularly for large datasets, images with high resolution, and complex models. This can impact the speed of model development and iteration and the feasibility of deploying real-time applications that require fast, efficient processing [13].

- This study critically evaluates the current landscape of deep learning and machine learning models used to classify tomato leaf diseases, highlighting a significant challenge in their deployment. As currently designed, the models are computationally heavy, characterized by high latency and overall inefficiency. Such attributes render them less than ideal for integration into real-world applications, particularly those aimed at supporting tomato cultivators through web-based and Android-based platforms [14].

3 Goal and Hypothesis

The present study leverages machine learning algorithms, achieving a maximal accuracy rate of approximately 95.58%. It posits that incorporating deep learning classification models, renowned for their superior feature extraction capabilities relative to traditional machine learning algorithms, could potentially elevate this accuracy beyond the current threshold. Nevertheless, the application of deep learning models is often constrained by their extensive training durations and significant latency, factors that could impede the utilization of large datasets for training purposes and practical deployment of real-life scenarios, especially beneficial to agricultural practitioners and framers. Therefore, this research endeavors to identify an optimal deep-learning framework alongside strategies to mitigate training time and latency, thereby facilitating real-world applicability.

The ultimate objective of this investigation is to develop a custom lightweight with low training time and low latency and an efficient deep-learning model that has achieved adequate classification accuracy and outperformed the majority of earlier studies on tomato leaf disease classification.

4 Experimentation

4.1 Dataset

To evaluate the proposed method, the PlantVillage tomato dataset plantdisease was utilized, comprising 10 distinct classes of tomato plant images, detailed as follows:

- **Bacterial Spot:** Small, water-soaked, angular to circular patches in dark brown, yellow, and black.
- **Early Blight:** Caused by Alternaria solani, with yellow patches evolving into concentric black rings.
- **Late Blight:** Stemming from Phytophthora infestans, marked by water-soaked lesions.
- **Leaf Mold:** Small, round, pale greenish-yellow patches on leaf tops, with clusters of brown dots.
- **Septoria Leaf Spot:** Caused by Septoria lycopersici, round spots with a yellow-haloed dot, appearing on the leaf undersides.
- **Spider Mites:**Leaves show a brown or yellow coating underneath.
- **Target Spot:** Caused by Corynespora cassiicola, small necrotic lesions with light brown centers and dark margins.
- **Mosaic Virus:**Results in dark and light spots in plant tissues.
- **Yellow Curl Virus:** Characterized by upwards and inwards rolling of leaf edges, yellowing leaflets, and smaller than usual leaves.
- **Healthy Leaves:** Images of leaves without any signs of disease.

This dataset, featuring nine classes representing various diseases and one class of healthy leaves, serves as the foundation for assessing the effectiveness of the advanced classification method.

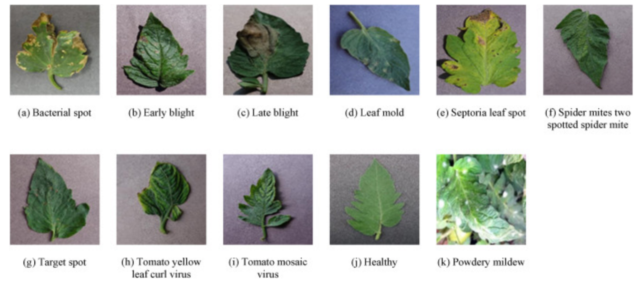

Figure 1: Sample images (random) for each class of tomato leaf in dataset

4.2 Data Pre-Processing:

4.2.1 Data Augmentation and Loading

- Images are loaded from the **'PlantVillage'** directory, converted to arrays using OpenCV, and resized to a default size of 75x75 pixels. This process involves iterating through each image file, reading the image, resizing it, and converting it to an array format.
- The images are labeled based on their folder names, which represent the class of the plant or disease.

4.2.2 Normalization and Splitting

- The pixel values of the images are normalized by dividing by 225.0 to scale them to a range [0, 1].
- The dataset is split into training and testing sets using a 80/20 split.

4.2.3 Label Encoding

- Labels are binarized using **LabelBinarizer**, converting categorical labels into a format suitable for model training (one-hot encoding).

4.3 Model Development

4.3.1 Implementing the MLP Model

Initially, our research endeavor embarked on the exploration of a relatively simpler deep neural network framework, specifically the Multilayer Perceptron (MLP) architecture, with the objective of employing it for image classification tasks. This decision was predicated on the aspiration that the MLP model would exceed the accuracy benchmarks established by contemporary machine learning models as documented by other scholars in the field [12].

- **Input Layer:** A Flatten layer is employed to convert 2-dimensional images into 1-dimensional vectors, facilitating their processing by the subsequent neural network layers. This layer consisted of 16875 neurons.

- **Hidden Layers:** The model incorporates five hidden layers with a descending configuration of neurons: starting with 1024 neurons and followed by two layers of 512 neurons each, then a layer with 256 neurons, and finally, a layer with 128 neurons. Each of these layers employs the ReLU (Rectified Linear Unit) activation function to introduce non-linearity, enhancing the network's ability to learn complex patterns. Additionally, BatchNormalization is applied after the ReLU activation in each hidden layer, aiming to normalize the layer inputs. This normalization is crucial for accelerating the training process and reducing sensitivity to network initialization parameters. Dropout techniques are further applied after batch normalization, with rates between 0.3 to 0.2, to mitigate overfitting by randomly nullifying a fraction of the inputs during training.

- **Output Layer:** For multi-class classification, the output layer is designed with a number of neurons corresponding to the number of target classes and utilizes a softmax activation function to generate probability distributions over the class labels.

- **Compilation:** The network undergoes compilation with two sets of loss functions and optimizers, ultimately applying binary crossentropy for the loss function and the Adam optimizer paired with an exponential decay schedule for the learning rate. Accuracy serves as the metric for model evaluation.

The model's training is conducted over 100 epochs with a batch size of 32. The procedure involves training on a designated dataset and validating the performance using a separate testing dataset. The learning rate is initially set at 10^{-3} and is adjusted throughout the training period according to a predefined schedule, to ensure efficient learning progression and convergence to optimal weights. This adaptive approach to the learning rate, alongside the model's comprehensive architecture and regularization techniques like Dropout and BatchNormalization, aims to achieve a delicate balance between learning efficiency and generalization capability, thereby enhancing model performance on unseen data. To improve model accuracy and robustness against overfitting.

The MLP model under investigation did not achieve the anticipated enhancement in accuracy, an objective set with the intention to surpass the benchmarks of existing machine learning models as delineated by other researchers [12]. Instead, the accuracy attained by our MLP model paralleled those reported in the current literature; a comprehensive dissection of these outcomes is furnished in the results section. To fulfill our aspiration of exceeding the accuracy levels of contemporary machine learning models highlighted by other scholars [12], it became apparent that a transition towards a more sophisticated deep learning architecture was required. Consequently, we explored alternative deep learning model architectures, aiming to achieve the objectives of this experimental study.

4.3.2 Implementing the CNN Model

The CNN model is more complex and suited for image data than the previously explored MLP model because it captures spatial hierarchies in the images. Convolutional Neural Networks (CNNs) represent specialized architectures explicitly designed for the task of image classification. They are proficient at identifying complex patterns using convolutional and pooling layers. The initial convolutional layers conduct operations where filters systematically move across input images, identifying unique characteristics such as edges, textures, and shapes. The following layers, known as pooling layers, perform downsampling of these features, effectively reducing the spatial dimensions while preserving critical information.

In this study, the CNN model features a multilayer architecture that is finely tuned to extract and process complex features within images autonomously. This model includes four convolutional layers, each equipped with filters of varying sizes and depths, allowing for the gradual identification of hierarchical patterns. This facilitates the representation of the input images in increasingly abstract forms. Additionally, including activation functions like the Rectified Linear Units (ReLU) between these layers introduces nonlinearities, thereby augmenting the network's ability to discern intricate relationships in the data.

Following the convolutional stages, our model incorporates two fully connected layers. Positioned towards the end of the network, these layers amalgamate the features extracted earlier, converting them into a form amenable to classification tasks. The neurons within these layers are densely interconnected, which supports the learning of more sophisticated abstractions, crucial for distinguishing among various classes of diseases featured in our dataset.

The culmination of our CNN model is marked by an output layer that employs a softmax activation function. This function calculates probabilities for each disease class, effectively translating the network's internal representations into actionable predictions. These predictions offer a probabilistic evaluation of the likelihood that the input images pertain to different disease categories within our dataset. The model is compiled using binary crossentropy as the loss function and Adam optimizer with an exponential decay learning rate schedule. This setup is aimed at optimizing the model for high accuracy in multi-label classification tasks. The comprehensive architecture and methodical progression of our model enable precise and dependable disease classification, an essential component in the accurate analysis and diagnosis of plant pathology.

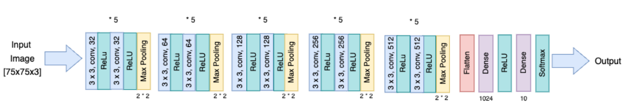

Figure 2: CNN Model Architecture

4.3.3 Hyperparameter

To ensure that the CNN is trained properly, we performed hyper parameter tuning in various combinations. After repeated tuning, the following parameters gave us the most effective results:

- **a. Learning Rate:** 1×10^{-3}
- **b. Optimization Algorithm:** Adam Optimizer
- **c. Number of Epochs:** 50
- **d. Batch-Size:** 32
- **e. Loss Function:** Binary Cross-Entropy
- **f. Decay type:** Exponential Decay
- **g. Decay steps:** 1000
- **h. Decay rate:** 0.9

In this study, we explore the efficacy and efficiency of Convolutional Neural Networks (CNNs) in achieving superior accuracy benchmarks compared to existing machine learning models. Our experiments yielded significant results, with CNNs surpassing current accuracy levels as detailed in the results section (Table 2). However, training these networks on a CPU—Apple M1 chip with an 8-core CPU and 16GB RAM—resulted in considerably longer durations. This increase in training time is attributable to the extensive computational demands of processing large datasets through the multiple layers of a CNN, including convolution, pooling, and fully connected layers. The depth and complexity of CNN architectures further exacerbate this, as deeper networks with more parameters and layers require numerous iterations over the dataset for optimization through backpropagation and gradient descent methods. Additionally, the process of hyperparameter tuning adds to the complexity and duration of model training.

To address the goal of reducing training time, we subsequently employed GPUs (specifically, a T4 GPU with 16GB of GDDR6 memory and 2,560 CUDA cores) with the anticipation of shortening this duration. Indeed, utilizing GPUs markedly decreased the training time, as outlined in the results section (Table 3). Nevertheless, this solution encountered a significant drawback in relation to the study's ultimate objective of facilitating real-time applications. Due to their architectural complexity and the substantial computational resources required for both training and inference, CNNs are considered heavyweight models. This complexity arises from their depth, comprising numerous convolutional, pooling, and fully connected layers with a significant parameter count. The computationally intensive nature of convolution operations, especially when processing high-resolution images or aiming for real-time performance, presents a notable challenge for deploying CNNs in applications demanding rapid processing, such as autonomous driving, video surveillance, and real-time language translation.

Moreover, the real-time deployment of CNN models poses stringent latency requirements, necessitating predictions within milliseconds to be viable. The inherent computational complexity of CNNs often results in longer inference times, particularly on hardware with limited processing capabilities like mobile devices. Despite the remarkable accuracy and transformative impact of CNNs in fields such as computer vision, their heavyweight nature and the associated computational challenges complicate their use in real-time environments.

Given these challenges, achieving the study's goal of real-time application feasibility necessitates the exploration of alternative models. These alternatives should be lightweight and exhibit lower latency to align with the real-time operational requirements, ultimately serving practical purposes such as enhancing agricultural practices for farmers. This redirection emphasizes the need for a balanced approach that considers both the computational efficiency and the accuracy of machine learning models in real-world applications.

4.4 Employing Transfer Learning

Transfer learning has become a cornerstone technique for enhancing the efficiency and practicality of deploying image classification systems in real-world scenarios. Transfer learning facilitates a considerable decrease in training duration and computational demands. This strategy employs a two-step process: initially leveraging a model trained on a vast, generic dataset to learn general features, followed by fine-tuning this model on a smaller, specific dataset tailored to the task at hand. This process not only expedites the training phase but also minimizes the requirement for computational resources, offering a distinct advantage for applications constrained by hardware limitations. Additionally, given that these models have already assimilated a broad spectrum of features, minimal adjustments are necessary to achieve precise predictions, thereby enabling quicker inference times essential for real-time applications. Hence, transfer learning models strike an optimal balance between accuracy and operational efficiency, rendering them exceptionally suitable for image classification tasks where rapid and accurate processing is paramount.

The methodology of transfer learning can be conceptualized as the application of knowledge acquired from addressing one problem to a different, albeit related, problem. It enables the repurposing of previously learned features or representations from one context to address challenges in a new dataset or task. This approach not only expedites the learning process but also mitigates the dependence on vast volumes of labeled data, often resulting in improved performance in the novel domain or task. Our application of transfer learning encompasses four distinct phases:

- **A. Pre-training:** The model is initialized with weights from pre-trained networks, harnessing the knowledge derived from a comprehensive dataset.
- **B. Training from Scratch:** This phase involves training the model from its initial state, without leveraging any pre-existing weights or acquired knowledge.
- **C. Linear Probing:** Adapting the pre-trained model's linear layers for the specific domain without modifying the convolutional base.
- **D. Fine-Tuning:** The pre-trained model undergoes a refinement process, permitting adjustments within deeper layers, particularly the convolutional layers, to accommodate the characteristics and features of the new dataset.

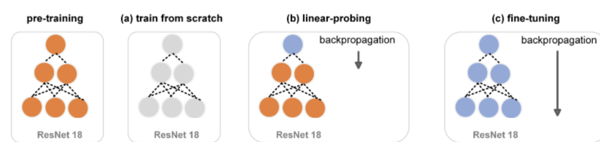

Figure 3: Transfer-learning nodal architecture

This structured approach to transfer learning underlines its effectiveness in reducing training time, conserving computational resources, and ensuring the rapid deployment of image classification systems in environments where swift and accurate decision-making is critical.

4.4.1 ResNet18 Architecture

ResNet18's architecture, characterized by its 18 weight layers and innovative residual connections, holds substantial significance. Leaf images, often intricate and nuanced, require models capable of discerning subtle variations to accurately identify diseases.

The design's 18 weight layers facilitate a more profound analysis of tomato leaf images. These layers enable the network to capture hierarchical features, ranging from basic edges and textures to more complex disease-related patterns. In the context of disease detection, these layers empower ResNet18 to extract and comprehend intricate details such as spots, discolorations, or deformations on tomato leaves, critical indicators of various diseases affecting the plants. However, what truly sets ResNet18 apart in this domain is its integration of residual connections. These connections mitigate the vanishing gradient problem encountered in deep networks, allowing for more efficient learning of intricate patterns within the leaf images.

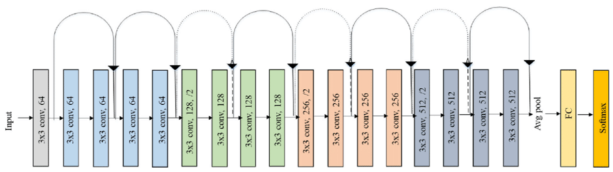

Figure 4: ResNet18 Model Architecture

In tomato leaf disease detection, where accurate identification of subtle disease-related features is crucial, the ability to train deeper networks without losing crucial information through these residual connections significantly enhances the model's capability. It enables the network to focus on learning specific disease-related nuances without being hindered by degradation issues, ultimately improving the effectiveness and efficiency of disease classification within the tomato leaf dataset.

Training the ResNet18 model on a Central Processing Unit (CPU)—specifically, an Apple M1 chip featuring an 8-core CPU and 16GB RAM—resulted in notably prolonged training periods. This significant increase in training duration can be attributed to the complex architecture and computational demands of the ResNet model. The ResNet18 architecture encompasses a multitude of convolutional layers, batch normalization processes, activation functions, and skip connections. Each of these elements plays a critical role in the model's capacity to assimilate extensive datasets effectively. However, CPUs are traditionally tailored for a broad range of general-purpose tasks, which inherently limits their ability to perform the parallel processing operations that are crucial for deep learning.

In pursuit of reducing the extensive training times, we transitioned to utilizing Graphics Processing Units (GPUs), specifically opting for a T4 GPU equipped with 16GB of GDDR6 memory and 2,560 CUDA cores. This strategic choice was driven by the GPU's architectural design, which is intrinsically optimized for parallel processing. GPUs excel in handling the matrix and vector computations that are ubiquitous in deep learning tasks, offering significantly greater throughput than CPUs. This adaptation was anticipated to markedly diminish the duration required to train the ResNet18 model, leveraging the GPU's superior computational efficiency and parallel processing capabilities.

The utilization of transfer learning with ResNet18 for tomato leaf disease detection aligns seamlessly with the objectives of this study, which prioritize model efficacy and operational efficiency. This approach achieves the study's aim to optimize training efficiency, latency and minimize computational demands while maintaining high diagnostic accuracy.

5 Experiments & Results

5.1 Evaluation Metrics

a) Accuracy: This measures the overall correctness of the model and is calculated as

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + RN)}$$

b) Precision: Precision indicates the accuracy of positive predictions and is defined as

$$Precision = \frac{TP}{(TP + FP)}$$

c) Recall : Recall assesses the model's ability to identify all relevant cases and is calculated

$$Recall = \frac{TP}{(TP + FN)}$$

d) F1-score: The F1-score is the harmonic mean of precision and recall, balancing the two. It is computed as

$$F1 = \frac{2 * TP}{(2 * TP + FP + FN)}$$

5.2 Results

In the initial phase of our research, we applied a simpler Multilayer Perceptron (MLP) architecture to the tomato dataset, obtaining an accuracy rate of 83.00%. This performance metric positioned our results within a similar range to those reported by other scholars for different machine learning models, as depicted in Fig5, Fig6, and Table1.

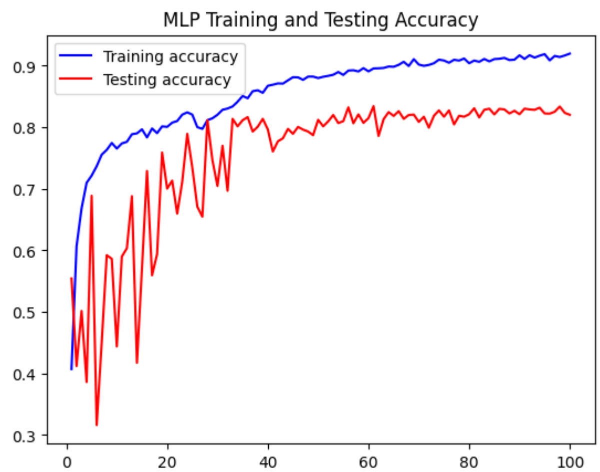

Figure 5: MLP Training and Testing Accuracy

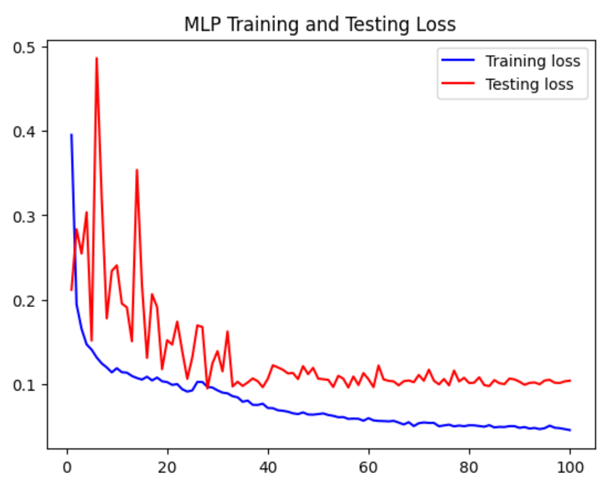

Figure 6: MLP Training and Testing Loss

Class	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Bacterial Spot	0.94	0.77	0.93	0.85
Early Blight	0.48	0.65	0.48	0.55
Late Blight	0.75	0.86	0.75	0.80
Leaf Mold	0.76	0.83	0.76	0.79
Septoria Leaf Spot	0.84	0.87	0.84	0.85
Spider Mites	0.82	0.77	0.82	0.79
Target Spot	0.77	0.70	0.77	0.74
Mosaic Virus	0.91	0.91	0.91	0.91
Yellow Curl Virus	0.94	0.83	0.94	0.88
Healthy Leaves	0.96	0.89	0.86	0.87
Overall	0.83	0.83	0.83	0.82

Table 1: MLP Evaluation Metrics Result

Subsequent to the realization that the MLP architecture did not meet the objectives of this study, our efforts pivoted towards the development of a more complex model. In this vein, we constructed a Convolutional Neural Network (CNN) with four convolution layers. The employment of this architecture yielded a significant increase in performance, achieving an accuracy rate of 97.88%. This marked a notable advancement beyond the accuracy levels reported for previously examined machine learning models. Fig7, Fig8, and Table2 respectively provide insight into the training and test accuracy, training and test loss, and the comprehensive evaluation criteria applied in the assessment of this model.

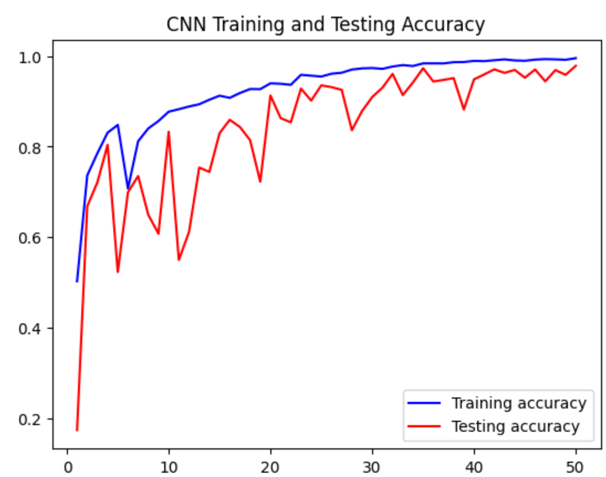

Figure 7: CNN Training and Testing Accuracy

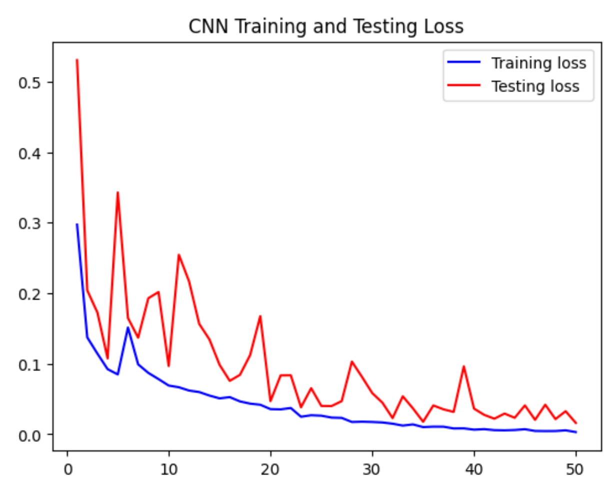

Figure 8: CNN Training and Testing Loss

Class	Accuracy	Precision	Recall	F1-Score
Bacterial Spot	0.99	1.00	0.99	0.99
Early Blight	0.91	0.94	0.91	0.92
Late Blight	0.98	0.96	0.98	0.97
Leaf Mold	0.99	0.97	0.99	0.98
Septoria Leaf Spot	0.97	0.98	0.97	0.97
Spider Mites	0.97	0.97	0.97	0.97
Target Spot	0.95	0.94	0.95	0.95
Mosaic Virus	0.99	1.00	0.99	1.00
Yellow Curl Virus	1.00	1.00	1.00	1.00
Healthy Leaves	1.00	0.99	1.00	1.00
Overall	0.98	0.98	0.98	0.98

Table 2: CNN Evaluation Metrics Result

Table3 presents a comparative analysis of the training durations for the CNN model on both CPU and GPU platforms, highlighting a substantial decrease in training time when utilizing the GPU. This observation underscores the efficacy of GPUs in expediting the training process for complex models like CNNs, thus offering a pragmatic solution to the computational challenges faced in deep learning tasks, such as handing large datasets.

Model	CPU Training Time	GPU Training Time
CNN	183m 52s	38m 7s
ResNet18	153m 30s	10m 7s

Table 3: Training Time using CPU and GPU

Upon a thorough evaluation of our designed CNN model, several limitations were identified, particularly concerning its applicability in real-time scenarios. These limitations prompted a strategic redirection towards developing a model characterized by reduced computational complexity and lower latency, without compromising the efficacy essential for high-accuracy image classification tasks. To this end, the ResNet18 architecture was selected as the focal point of our ongoing research efforts. Our work involves constructing and refining the ResNet18 model, with a specific focus on identifying and optimizing the model's hyperparameters, which is fine-tune the ResNet18 model.

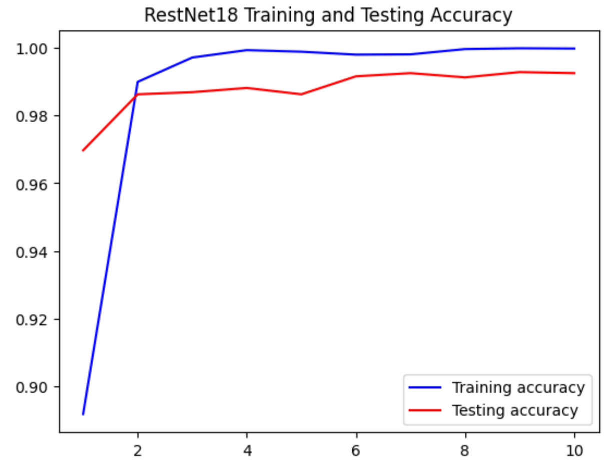

Figure 9: ResNet18 Training and Testing Accuracy

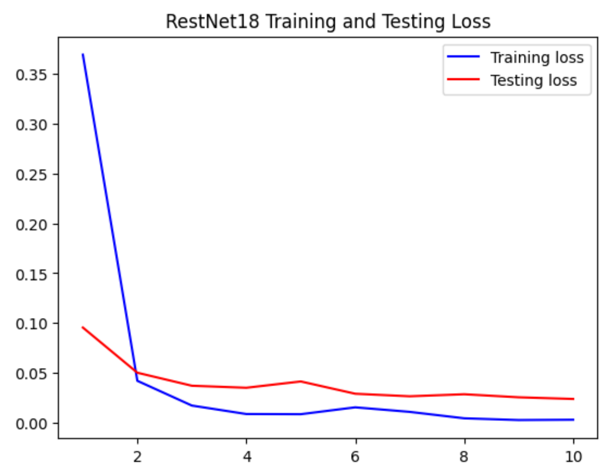

Figure 10: ResNet18 Training and Testing Loss

Class	Accuracy	Precision	Recall	F1-Score
Bacterial Spot	1.00	1.00	1.00	1.00
Early Blight	0.94	0.98	0.94	0.96
Late Blight	0.98	0.99	0.98	0.98
Leaf Mold	0.99	0.99	0.99	0.99
Septoria Leaf Spot	0.99	0.98	0.99	0.99
Spider Mites	1.00	0.99	1.00	0.99
Target Spot	0.99	0.99	0.99	0.99
Mosaic Virus	1.00	1.00	1.00	1.00
Yellow Curl Virus	1.00	0.98	1.00	0.99
Healthy Leaves	1.00	1.00	1.00	1.00
Overall	0.99	0.99	0.99	0.99

Table 4: ResNet18 Evaluation Metrics Result

The adoption of the ResNet18 architecture resulted in a enhancement in model performance, achieving an accuracy rate of 99.41%. This represents a marked improvement over the accuracy levels previously achieved by other classical machine learning models, including MLP and CNN deep learning networks explored in this study. Figures 9 and 10, along with Table 4, provide detailed insights into the training and testing accuracy, training and testing loss, and the comprehensive evaluation metrics used to assess this model. These results underscore the superior capability of the ResNet18 architecture in handling complex image classification tasks, such as tomato leaf disease detection, and highlight its potential as a highly effective tool in the realm of agricultural technology.

Table 3 presents a comparative analysis of the training durations for the ResNet18 model on both CPU and GPU platforms. It highlights a significant reduction in training time when utilizing the GPU, an observation that underscores the efficacy of GPUs in accelerating the training process for complex models like ResNet18. This advantage is particularly valuable in addressing the computational challenges inherent in deep learning tasks, such as managing large datasets. The use of GPUs offers a practical solution by enhancing processing speeds, thereby facilitating more efficient model development and iteration crucial for advanced deep learning applications.

To meet the objectives of this study, specifically to develop a model suitable for real-time detection, we conducted a comprehensive evaluation of the computational efficiency of the MLP, standard CNN, and ResNet18 models. Latency measurements were carried out under controlled testing conditions, utilizing the same dataset across all models to ensure a consistent basis for comparison. This methodology involved recording each model's time required to process inputs and generate outputs, thus quantifying their response times. This systematic assessment allowed for an objective comparison of the model's performance in terms of speed and efficiency, which are critical factors for their applicability in real-time applications and the major objective of this study.

The latency results, detailed in Table 5, highlight a notable performance disparity between the models tested. The ResNet18 model exhibited better processing speeds, completing tasks in merely 1.10 seconds, in contrast to the traditional CNN model, which required 3.23 seconds. This enhanced speed stems from ResNet18's architectural efficiencies, including residual connections that prevent the vanishing gradient problem and accelerate both training and inference. Additionally, ResNet18's design minimizes unnecessary computations, incorporates batch normalization to stabilize activations and speed up convergence, reuses learned features to reduce computational demands, and occupies less memory due to its smaller size (44MB). Also this difference emphasizes the enhanced efficiency of the ResNet18 architecture, making it particularly well-suited for applications that require rapid data processing and prompt decision-making for tomato leaf disease detection. Such attributes make ResNet18 a valuable asset in scenarios where it can be deployed for real-life applications.

Time	**CNN**	**ResNet18**
Latency	3s 23ms	1s 10ms

Table 5: Latency using CNN and ResNet18

5.3 Conclusions

We initially anticipated significant improvements in accuracy by transitioning from classical machine learning approaches to deep learning models. However, our baseline model, a Multilayer Perceptron (MLP) architecture, did not achieve higher accuracy compared to the classical machine learning techniques reported in previous studies by other scholars. Consequently, the MLP architecture did not fulfill the accuracy objectives of this study, prompting a reevaluation of our modeling strategy to better align with our goals of enhancing diagnostic precision in plant disease detection. As result, we expanded our exploration from the Multilayer Perceptron (MLP) architecture to Convolutional Neural Networks (CNNs). Training CNN architectures from scratch yielded significant improvements in accuracy over both classical machine learning techniques and the MLP architecture previously employed in this study due to the inherent capabilities of CNNs. Upon achieving considerable accuracy with the CNN model, our next objective was to reduce training time to facilitate the model's future application on large datasets in real-life scenarios. We successfully addressed this challenge by utilizing GPUs, which, as detailed in the results section above, significantly reduced the training time for our CNN model.

However, despite its high accuracy, the CNN model we just designed proved to be a heavier model with higher latency, the size is about 64MB which made it unsuitable for our ultimate goal of developing a model feasible for real-time detection.

This prompted further exploration into transfer learning. After several tests and trials with different models, we found that the ResNet18 architecture met our needs ideally. The size of ResNet18 is about 44MB, which is lighter than our CNN model. ResNet18 not only achieved high accuracy but also offered a balance between computational efficiency and performance, making it well-suited for real-time applications and thus aligning with the primary objectives of our study. The results achieved with the ResNet18 model, which reached an impressive 99.31% accuracy, exceeded even our most optimistic expectations. This exceptional outcome not only confirms the efficacy of transfer learning but also sets new benchmarks in the field of plant pathology, reshaping our standards of what is achievable.

To achieve the study's objectives, we focused on reducing the training time for the ResNet18 model, which was successfully accomplished through the utilization of GPUs. This adaptation significantly decreased the training duration, making it feasible to train the model on large datasets in future applications. Further, we evaluated the latency performance of ResNet18 compared to the standard CNN model and observed a notable reduction in latency, affirming the suitability of the ResNet18 model for real-time applications.

In conclusion, this study successfully validated the hypothesis by achieving all set objectives by applying transfer learning to the ResNet18 model, providing a scalable and efficient solution for detecting tomato leaf diseases. The ResNet18 model demonstrated exceptional capabilities, being lightweight with reduced training time and low latency while maintaining high classification accuracy. It outperformed most classical machine learning models and surpassed some of the deep learning models explored in this study. The superior performance of the ResNet18 model, combined with its low latency and lightweight nature, makes it highly feasible for real-time deployment.

Furthermore, the use of GPUs as computational resources was instrumental in reducing training times, enabling the model to be scalable and adaptable for training on extensive datasets to accommodate broader applications. These results carry profound implications for the field of plant disease detection, showcasing the transformative impact of integrating transfer learning with CNN-based diagnostics. This strategy significantly reduces the reliance on large datasets and enhances the adaptability of the models to specialized agricultural tasks, setting a new standard in the application of deep learning methodologies for efficient and precise image analysis in agriculture.

5.4 Future Works

This work can further be extended in the following ways:

- One immediate area of interest is the exploration of additional advanced transfer learning architectures beyond ResNet18, such as InceptionV3 or EfficientNet, which may offer different advantages in terms of accuracy, efficiency, and scalability. Investigating these models could provide insights into the trade-offs between model complexity and real-time performance capabilities.

- To further enhance the effectiveness of the Convolutional Neural Network (CNN) model, it is proposed to employ Conditional Generative Adversarial Networks (C-GANs) prior to the training phase. C-GANs can be utilized to artificially augment the dataset by generating a more diverse array of image data. This method would allow the CNN to train on a broader spectrum of variations in plant disease appearances, potentially capturing subtler, less common features that are not adequately represented in the original dataset. By enriching the dataset in this manner, the model can achieve greater generalization capabilities, thus improving its robustness and accuracy in identifying a wider range of disease states across different tomato plant conditions.

- Exploring semi-supervised learning approaches presents a promising avenue for effectively utilizing unlabeled data to enhance limited labeled datasets. Techniques such as pseudo-labeling or self-training could be particularly advantageous in the context of plant pathology, where the abundance of unlabeled images remains largely untapped. By incorporating these semi-supervised methods, models can extend their learning beyond the confines of labeled datasets, allowing for the assimilation of valuable information from unlabeled images. This approach not only expands the training dataset at minimal additional cost but also improves the model's ability to generalize from a more comprehensive representation of data.

References

[1] P. Lindhout, W. Korta, M. Cislik, I. Vos and T. Gerlagh, "Further identification of races of cladosporium fulvum (Fulvia fulva) on tomato originating from The Netherlands France and Poland", *Netherlands J. Plant Pathol.*, vol. 95, pp. 143-148, May 1989.

[2] K. Kubota, S. Tsuda, A. Tamai and T. Meshi, "Tomato mosaic virus replication protein suppresses virus-targeted posttranscriptional gene silencing", *J. Virol.*, vol. 77, no. 20, pp. 11016-11026, Oct. 2003.

[3] M. Tian, B. Benedetti and S. Kamoun, "A second Kazal-like protease inhibitor from phytophthora infestans inhibits and interacts with the apoplastic pathogenesis-related protease P69B of tomato", *Plant Physiol.*, vol. 138, no. 3, pp. 1785-1793, Jul. 2005.

[4] R. Chaerani and R. E. Voorrips, "Tomato early blight (Alternaria solani): The pathogen genetics and breeding for resistance", *J. Gen. Plant Pathol.*, vol. 72, no. 6, pp. 335-347, Dec. 2006.

[5] W. Gao, B. Li, Y. Shi and X. Xie, "Studies on pathogenicity differentiation of Corynespora cassiicola isolates against cucumber tomato and eggplant", *Acta Horticulturae Sinica*, vol. 38, no. 3, pp. 465-470, 2011.

[6] A. M. Dickey, L. S. Osborne and C. L. Mckenzie, "Papaya (Carica papaya Brassicales: Caricaceae) is not a host plant of tomato yellow leaf curl virus (TYLCV; FamilyGeminiviridae GenusBegomovirus)", *Florida Entomologist*, vol. 95, no. 1, pp. 211-213, Mar. 2012.

[7] W. Matt et al., *Smallholders Food Security and the Environment*, Rome, Italy:International Fund for Agricultural Development, Mar. 2013.

[8] C. A. Harvey, Z. L. Rakotobe, N. S. Rao, R. Dave, H. Razafimahatratra, R. H. Rabarijohn, et al., "Extreme vulnerability of smallholder farmers to agricultural risks and climate change in madagascar", *Phil. Trans. Roy. Soc. B Biol. Sci.*, vol. 369, no. 1639, Apr. 2014.

[9] L. E. B. Blum, "Reduction of incidence and severity of septoria lycopersici leaf spot of tomato with bacteria and yeasts", *Ciência Rural*, vol. 30, no. 5, pp. 761-765, Oct. 2000.

[10] L. Zhang, F. Yang, Y. Daniel Zhang and Y. J. Zhu, "Road crack detection using deep convolutional neural network", *Proc. IEEE Int. Conf. Image Process. (ICIP)*, pp. 3708-3712, Sep. 2016.

[11] S. P. Jia, H. J. Gao and X. Hang, "Advances in image recognition technology of crop diseases and insect pests based on deep learning", *Trans. Chin. Soc. Agricult. Mach.*, vol. 50, pp. 314-317, Jul. 2019.

[12] Javidan, S. M., Banakar, A., Vakilian, K. A., & Ampatzidis, Y. (2023). "Tomato leaf diseases classification using image processing and weighted ensemble learning", *Agronomy Journal*, 00, 1–21. `https://doi.org/10.1002/agj2.21293`

[13] E. BUBER and B. DIRI, "Performance Analysis and CPU vs GPU Comparison for Deep Learning," 2018 6th International Conference on Control Engineering & Information Technology (CEIT), Istanbul, Turkey, 2018, pp. 1-6, doi: 10.1109/CEIT.2018.8751930.

[14] S. G. Paul et al., "A Real-Time Application-Based Convolutional Neural Network Approach for Tomato Leaf Disease Classification," in *Array* (New York), vol. 19, no. 100313, 2023. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S2590005620300313`