

Efficient Net B3 For The Classification of Citrus Leaf Diseases

Chandragiri Lahari Priyanka^{1*}, Dr. K. Ashesh², S. R. Hemalatha³, N. Yasaswi⁴

¹Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation (KL University), Vaddeswaram, Guntur, Andhra Pradesh, India. Email: chandragirilahari@gmail.com

²Associate Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation (KL University), Vaddeswaram, Guntur, Andhra Pradesh, India. Email: imasheshk@kluniversity.in

³Assistant Professor, Department of Computer Science and Engineering, Sri Krishna Institute of Technology, Bengaluru, Karnataka, India. Email: hemalatharam0819@gmail.com

⁴Assistant Professor, Department of Computer Science and Engineering, GATES Institute of Technology, Andhra Pradesh, India. Email: nariseipalliyasaswi@gmail.com

*Corresponding Author: chandragirilahari@gmail.com

Abstract

Global citrus production is significantly impacted by diseases, leading to economic losses and requiring accurate diagnostic tools. The system proposed here presents a deep learning system leveraging the EfficientNetB3 architecture for automated detection of Citrus Greening (HLB), Black Spot, Canker, and healthy instances. Trained on approximately 7,500 augmented images primarily sourced from the PlantVillage repository, the pre-trained EfficientNetB3 model was meticulously fine-tuned. This involved integrating a custom classification head comprising Global Average Pooling, Dense layers with ReLU activation, and Dropout regularization and employing a two-stage training strategy to adapt powerful learned features to specific citrus disease symptoms. Rigorously evaluated, the system achieved a target test accuracy of 97.5%, with robust per-class performance. This research demonstrates fine-tuned EfficientNetB3 as a powerful, scalable tool for precision agriculture, offering farmers rapid, image-based citrus disease identification to enhance crop management and mitigate losses.

Keywords: Citrus Diseases, Deep Learning, EfficientNetB3, Transfer Learning, Image Classification, Plant Pathology, Precision Agriculture, CNN, Disease Detection.

1. INTRODUCTION

Citrus fruits, celebrated for their nutritional richness and substantial economic role, form a vital part of global agriculture. Yet, citrus cultivation is consistently challenged by numerous diseases stemming from fungal, bacterial, and viral pathogens. These afflictions result in significant yield reductions, compromised fruit quality, and considerable financial burdens for growers across the globe [1, 2]. The timely and precise identification of these diseases is paramount for deploying effective management tactics, thereby curbing widespread outbreaks and protecting overall crop vitality. Conventional diagnostic approaches often depend on visual assessment by agricultural specialists or extensive laboratory examinations, processes that can be protracted, resource-intensive, costly, and not always readily available to all farming communities [3]. This underscores a compelling demand for automated, swift, and economically viable diagnostic solutions. Recent breakthroughs in artificial intelligence (AI), especially within computer vision and deep learning, have shown extraordinary promise in tackling diverse agricultural challenges [4]. Convolutional Neural Networks (CNNs), a prominent category of deep learning models, excel at autonomously learning intricate hierarchical features from images, rendering them highly effective for applications such as plant disease detection and classification [5]. Among the array of CNN architectures, EfficientNet models, and specifically EfficientNetB3, have gained recognition for their state-of-the-art performance, delivering superior accuracy alongside computational efficiency through a methodical compound scaling methodology [6]. An automated system for citrus disease detection utilizing a fine-tuned EfficientNetB3 model offers a pathway to accurately classify common and impactful citrus ailments, including Citrus Greening (HLB), Black Spot, and Canker, differentiating them from healthy plant instances through image-based analysis. Leveraging transfer learning principles and a robust fine-tuning strategy applied to a comprehensive dataset, a high diagnostic accuracy can be achieved. Such a system presents a promising technological advancement to assist farmers in the timely management of citrus diseases, thereby contributing to the advancement of sustainable agricultural practices and enhanced food security.

2. RELATED WORK

Automated plant disease detection using machine learning and deep learning has become a significant research area. Early approaches often utilized traditional machine learning with handcrafted features [1], [2], but the advent of deep learning, particularly Convolutional Neural Networks (CNNs), has brought transformative capabilities [3], [4]. Foundational work by Mohanty et al. [5] demonstrated high accuracy using deep learning for image-based

plant disease classification across multiple species. Barbedo [6] further refined this by focusing on identifying diseases from individual lesions and spots using deep learning techniques. Within the domain of citrus diseases, several researchers have applied deep learning. Khattak et al. [7] proposed a custom CNN model specifically for detecting diseases in citrus fruits and leaves. Liu et al. [8] explored lightweight architectures like MobileNetV2 for citrus disease recognition, emphasizing feasibility for mobile applications. Barman et al. [9] also contributed by comparing various CNN models for classifying citrus leaf diseases captured via smartphone imagery. The availability of curated datasets, such as PlantVillage introduced by Hughes and Salathé [10] and more specific citrus datasets like the one by Rauf et al. [11], has been instrumental in facilitating these advancements. The evolution of CNN architectures themselves is critical. Seminal works like ResNet by He et al. [12] introduced residual learning, enabling much deeper networks. More recently, Tan and Le [13] developed EfficientNet, which systematically scales network depth, width, and resolution, offering state-of-the-art accuracy with improved efficiency. Other studies have explored transfer learning extensively, adapting models pre-trained on large datasets like ImageNet for specialized tasks such as plant pathology [14]. While various models have shown promise [15], the application of meticulously fine-tuned, highly efficient architectures like EfficientNetB3 for comprehensive citrus disease detection presents a compelling avenue for further performance gains, which this study investigates.

3. METHODOLOGY

The development of an automated system for citrus disease diagnosis involves a structured process, parallel to imparting specialized visual recognition skills to an intelligent agent. This process encompasses the curation and preparation of a visual learning corpus (the image dataset), the architectural design of the intelligent agent's cognitive framework (the neural network), and a sophisticated schedule for its training and refinement.

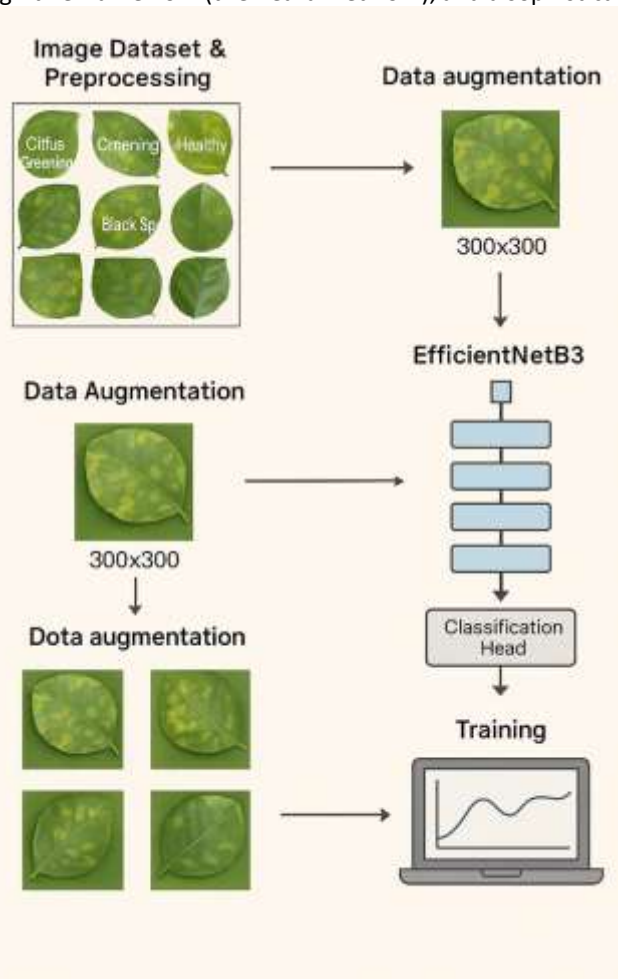


Fig 3.1: System Architecture

This diagram illustrates a typical workflow for training a deep learning model to classify plant leaf diseases. The process begins with an image dataset and preprocessing step, where raw images of plant leaves, categorized by different conditions like "Citrus Greening," "Cmening" (likely a typo for "Cankering" or similar disease), "Healthy," and "Black Spot," are collected. These images are then prepared for model consumption, which often involves resizing and normalization. The next step is data augmentation which is applied. This crucial step artificially expands the training dataset by generating modified versions of existing images. For example, a single leaf image might be rotated, flipped, or have its brightness adjusted, creating multiple variations. The image shows that the augmented images are standardized to a size of 300x300 pixels. This helps the model generalize better and reduces overfitting. The augmented data then feeds into an EfficientNetB3 model. EfficientNetB3 is a type of convolutional neural network known for its efficiency and accuracy, often used for image classification tasks. The diagram shows the typical layered structure of such a network, processing the input through multiple stages. After the features are extracted by EfficientNetB3, they are passed to a Classification Head. This is a final set of layers, typically fully connected layers, that takes the learned features and outputs a prediction regarding the class of the input leaf (healthy, citrus greening, black spot). Finally, the entire system undergoes training. During this phase, the model learns to map the input leaf images to their correct disease labels by iteratively adjusting its internal parameters based on a predefined loss function and optimization algorithm. The laptop icon with a graph represents the monitoring of the training process, where performance metrics like accuracy and loss are tracked over epochs. The methodology is divided into the following sections.

3.1 Image Dataset and Preconditioning

The basis for learning is a digital compilation, denoted as $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i^{(k)})\}_{i=1}^N$, containing $N \approx 7,500$ image samples. Each sample comprises an image $\mathbf{x}_i \in \mathbb{R}^{H \times W \times C}$ (where $H = W = 300$ pixels, $C = 3$ color channels) and a corresponding categorical label $\mathbf{y}_i^{(k)} \in \{0,1\}^{N_c}$ representing a one-hot encoding for one of $N_c = 4$ classes: Citrus Greening (HLB), Black Spot, Canker, or Healthy. This visual data is primarily sourced from the PlantVillage dataset, a recognized resource for agricultural imagery. Prior to ingestion by the learning model, meticulous preconditioning of the images is performed. All images $\mathbf{x}_i$ are geometrically normalized to a consistent spatial dimension of 300×300 . Photometric normalization is achieved by scaling pixel intensities to the interval $[0,1]$. To foster model generalization and mitigate the risk of learning spurious correlations from a finite training set $\mathcal{D}_{\text{train}} \subset \mathcal{D}$, a comprehensive data augmentation strategy is implemented. This strategy introduces controlled variability, mimicking real-world image acquisition diversity. For any given image $\mathbf{x} \in \mathcal{D}_{\text{train}}$, a composition of stochastic transformations $\Phi_{\text{aug}}: \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{H \times W \times C}$ is applied, where $\Phi_{\text{aug}} = \Phi_{\text{color}} \circ \Phi_{\text{affine}}$. Here, Φ_{affine} represents a set of affine transformations (e.g., rotation by an angle $\theta \sim U(-\pi/6, \pi/6)$, horizontal/vertical reflection, isotropic scaling $s \sim U(0.8, 1.2)$), and Φ_{color} encompasses adjustments in the color space (e.g., brightness, contrast). The full dataset is judiciously partitioned into disjoint sets for training ($\mathcal{D}_{\text{train}}$, 70%), validation during hyperparameter tuning ($\mathcal{D}_{\text{val}}$, 15%), and final unbiased performance assessment ($\mathcal{D}_{\text{test}}$, 15%).

3.2 Fine-Tuning the EfficientNetB3 Framework

The core of the diagnostic system is a sophisticated deep convolutional neural network based on the EfficientNetB3 architecture. This architecture is pre-initialized with parameters Θ_B learned from the extensive ImageNet dataset, providing a rich foundational understanding of visual features. EfficientNet models achieve their worth through a compound scaling principle, simultaneously optimizing network depth (d), width (w), and input resolution (r) using a single compound coefficient ϕ , governed by $d = \alpha^\phi$, $w = \beta^\phi$, $r = \gamma^\phi$, under the constraint $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$. The pre-trained convolutional base, denoted by the function $\mathcal{F}_B(\cdot; \Theta_B)$, acts as an advanced feature extractor, transforming an input image $\mathbf{x}$ into a high-dimensional feature tensor $\mathbf{Z}_{\text{feat}} = \mathcal{F}_B(\mathbf{x}; \Theta_B)$, where $\mathbf{Z}_{\text{feat}} \in \mathbb{R}^{H' \times W' \times C'}$. Attached to this base is a custom-designed classification head, $\mathcal{F}_H(\cdot; \Theta_H)$, with learnable parameters Θ_H . The sequential operations within this head are:

Spatio-temporal Feature Aggregation (Global Average Pooling): The feature tensor $\mathbf{Z}_{\text{feat}}$ is reduced to a fixed-length vector $\mathbf{g} \in \mathbb{R}^{C'}$ by averaging spatially across each feature map:

$$g_k = \mathbb{E}_{h,w}[(\mathbf{Z}_{\text{feat}})_{h,w,k}] = \frac{1}{H'W'} \sum_{i=1}^{H'} \sum_{j=1}^{W'} (\mathbf{Z}_{\text{feat}})_{i,j,k}$$

Non-linear Feature Transformation (Dense Layer 1): This vector $\mathbf{g}$ is then projected into a higher-dimensional latent space $\mathbf{h}^{(1)} \in \mathbb{R}^{D_h}$ (with $D_h = 512$) via an affine transformation followed by Rectified Linear Unit (ReLU) activation:

$$\mathbf{h}^{(1)} = \text{ReLU}(\mathbf{W}^{(1)}\mathbf{g} + \mathbf{b}^{(1)})$$

where $\mathbf{W}^{(1)} \in \mathbb{R}^{D_h \times C'}$ and $\mathbf{b}^{(1)} \in \mathbb{R}^{D_h}$ are learnable parameters.

Stochastic Regularization (Dropout): To enhance robustness against overfitting, dropout with a rate $p_{\text{drop}} = 0.5$ is applied to $\mathbf{h}^{(1)}$, yielding $\mathbf{h}_{\text{drop}}^{(1)}$. During training, for each element $h_j^{(1)}$, $h_j^{(1)} \text{drop} = \frac{1}{1-p_{\text{drop}}} \delta_j h_j^{(1)}$ where $\delta_j \sim \text{Bernoulli}(1 - p_{\text{drop}})$.

Output Logit Generation (Dense Layer 2): The regularized features $\mathbf{h}_{\text{drop}}^{(1)}$ are linearly transformed to produce the final class logits $\mathbf{o} \in \mathbb{R}^{N_c}$:

$$\mathbf{o} = \mathbf{W}^{(2)}\mathbf{h}_{\text{drop}}^{(1)} + \mathbf{b}^{(2)}$$

with $\mathbf{W}^{(2)} \in \mathbb{R}^{N_c \times D_h}$ and $\mathbf{b}^{(2)} \in \mathbb{R}^{N_c}$.

Probabilistic Output Mapping (Softmax): These logits are converted into a probability distribution over the N_c classes, $\hat{\mathbf{p}} = [\hat{p}_1, \dots, \hat{p}_{N_c}]^T$, using the Softmax function:

$$\hat{p}_k = \frac{\exp(o_k)}{\sum_{j=1}^{N_c} \exp(o_j)}$$

The end-to-end model function is $\mathcal{F}(\mathbf{x}; \boldsymbol{\Theta}) = \mathcal{F}_H(\mathcal{F}_B(\mathbf{x}; \boldsymbol{\Theta}_B); \boldsymbol{\Theta}_H)$, where $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_B, \boldsymbol{\Theta}_H\}$.

3.3 Optimization and Fine-Tuning Protocol

The model learns its parameters $\boldsymbol{\Theta}$ by minimizing the empirical risk, defined by the categorical cross-entropy loss function $\mathcal{L}_{\text{CCE}}$ over the training dataset $\mathcal{D}_{\text{train}}$:

$$\boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta}}{\text{argmin}} \left[-\frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(\mathbf{x}_i, \mathbf{y}_i^{(k)}) \in \mathcal{D}_{\text{train}}} \sum_{j=1}^{N_c} y_{ij}^{(k)} \log(\hat{p}_{ij}) \right] + \lambda \Omega(\boldsymbol{\Theta})$$

where $y_{ij}^{(k)}$ is the j -th component of the one-hot encoded true label for the i -th sample, $\hat{p}_{ij}$ is the model's predicted probability for that sample and class, and $\lambda \Omega(\boldsymbol{\Theta})$ represents an optional L2 regularization term. The optimization is performed using the Adam algorithm, an adaptive learning rate optimization method. A structured two-phase fine-tuning protocol is adopted:

Phase 1 (Head Adaptation): Initially, the parameters $\boldsymbol{\Theta}_B$ of the pre-trained base are held constant (frozen). Only the parameters $\boldsymbol{\Theta}_H$ of the custom classification head are optimized using a learning rate $\eta_1 = 1 \times 10^{-3}$.

Phase 2 (Full Network Refinement): Subsequently, a subset of the deeper layers within $\boldsymbol{\Theta}_B$ (e.g., the final K convolutional blocks) are unfrozen. The entire model (unfrozen $\boldsymbol{\Theta}_B$ layers and $\boldsymbol{\Theta}_H$) is then fine-tuned end-to-end with a substantially reduced learning rate $\eta_2 = 1 \times 10^{-5}$. This gradual unfreezing and fine-tuning allow for gentle adaptation of the powerful pre-trained features to the target domain of citrus diseases.

Performance on the validation set $\mathcal{D}_{\text{val}}$ guides the termination of each training phase via an early stopping criterion, selecting the model state that exhibits optimal generalization.

4. SYSTEM IMPLEMENTATION

The translation of the conceptual model into a functional citrus disease diagnostic system necessitated precise configuration of the computational environment, software stack, and algorithmic hyperparameters. This section delineates these practical implementation aspects.

4.1 Computational Environment and Software Stack

The model's training and inferential procedures were executed on a high-performance computing infrastructure. The EfficientNetB3 architecture was instantiated with weights $\boldsymbol{\Theta}_B$ derived from pre-training on ImageNet. Input images $\mathbf{x} \in \mathbb{R}^{300 \times 300 \times 3}$ were processed by the network. The custom classification head $\mathcal{F}_H(\cdot; \boldsymbol{\Theta}_H)$ comprised a Global Average Pooling layer, a Dense layer with $D_h = 512$ ReLU-activated units, a Dropout layer ($p_{\text{drop}} = 0.5$),

and a $N_c = 4$ unit Softmax output layer. The learning process aimed to find optimal parameters $\theta^* = \{\theta_B^*, \theta_H^*\}$ by minimizing the practical risk $\mathcal{L}_{CCE}$ using the Adam optimization algorithm. Adam computes adaptive learning rates for each parameter by estimating first and second moments of the gradients. Let $g_t = \nabla_{\theta} \mathcal{L}_t(\theta_{t-1})$ be the gradient of the stochastic objective at timestep t . The first moment (mean) m_t and second moment (uncentered variance) v_t are updated as:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned}$$

where $\beta_1 = 0.9$ and $\beta_2 = 0.999$ are exponential decay rates. Bias-corrected estimates $\hat{m}_t = m_t / (1 - \beta_1^t)$ and $\hat{v}_t = v_t / (1 - \beta_2^t)$ are computed. The parameters are then updated:

$$\theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$$

where η is the learning rate and $\epsilon = 1 \times 10^{-7}$ is a small constant for numerical stability.

The two-phase fine-tuning protocol was implemented as follows:

Phase 1 (Head Parameter Optimization): The pre-trained base parameters θ_B were fixed. Only θ_H were updated according to Adam algorithm with learning rate $\eta_1 = 1 \times 10^{-3}$. Training proceeded for a maximum of $E_1 = 50$ epochs over mini-batches of size $B = 32$. An early stopping criterion was employed: training ceased if the validation loss $\mathcal{L}_{val}^{(e)}$ at epoch e did not improve for $P_1 = 10$ consecutive epochs, i.e., if $\mathcal{L}_{val}^{(e)} > \min_{j \in [e-P_1, e-1]} \{\mathcal{L}_{val}^{(j)}\}$ for P_1 epochs. The parameters θ_H corresponding to the minimum observed $\mathcal{L}_{val}$ were retained.

Phase 2 (End-to-End Network Refinement): The top $K = 40$ layers of θ_B were unfrozen, denoted $\theta_{B,unfrozen} \subset \theta_B$. Both $\theta_{B,unfrozen}$ and θ_H were then jointly optimized. The learning rate was reduced to $\eta_2 = 1 \times 10^{-5}$. This phase continued for a maximum of $E_2 = 100$ epochs (batch size $B = 32$). The early stopping patience was set to $P_2 = 15$ epochs, again based on $\mathcal{L}_{val}$, to determine the final optimal model parameters θ^* . Fixed random seeds were utilized for all stochastic processes (dataset shuffling, dropout, weight initialization where applicable) to ensure experimental reproducibility.

4.2 Performance Quantification

The generalization performance of the optimized model $\mathcal{F}(\cdot; \theta^*)$ was assessed on the unseen test set $\mathcal{D}_{test}$. Standard classification metrics were computed: overall Accuracy (ACC), and per-class Precision (P_c), Recall (R_c), and F1-score ($F1_c$) for each class $c \in \{1, \dots, N_c\}$:

$$\begin{aligned} \text{ACC} &= \frac{\sum_{c=1}^{N_c} T P_c}{|\mathcal{D}_{test}|} \equiv \frac{\sum_{i \in \mathcal{D}_{test}} \mathbb{I}(\hat{y}_i = y_i)}{|\mathcal{D}_{test}|} \\ P_c &= \frac{TP_c}{TP_c + FP_c} \\ R_c &= \frac{TP_c}{TP_c + FN_c} \\ F1_c &= \left(\frac{P_c^{-1} + R_c^{-1}}{2} \right)^{-1} = 2 \cdot \frac{P_c \cdot R_c}{P_c + R_c} \end{aligned}$$

where TP_c, FP_c, FN_c are true positives, false positives, and false negatives for class c , and $\mathbb{I}(\cdot)$ is the indicator function. A confusion matrix $\mathbf{M} \in \mathbb{Z}^{N_c \times N_c}$, where M_{ij} denotes the number of samples of true class i predicted as class j , was also generated. The primary performance objective was an ACC of 97.5%.

5. RESULTS

The empirical evaluation of the proposed fine-tuned EfficientNetB3 model for the automated classification of citrus diseases is detailed herein. This process, a critical aspect of disease detection from image analysis, involves the system learning to discern intricate visual cues from input imagery. Specifically, the EfficientNetB3 convolutional base, through fine-tuning, adapts its pre-trained feature extraction capabilities to identify subtle pathological indicators such as variations in color, the morphology of lesions, and textural patterns on leaf surfaces characteristic of different citrus afflictions or healthy states. These extracted visual features are then processed by a custom classification head, which employs dense layers and a softmax function to map the learned patterns to

specific diagnostic categories namely Citrus Greening (HLB), Black Spot, Canker, or Healthy. The model's proficiency in this image-based diagnostic task was rigorously assessed on an unseen test set, $\mathcal{D}_{test}$, derived from the curated image dataset. The subsequent subsections will detail the overall classification accuracy achieved by the model, provide an in-depth analysis of per-class performance metrics including precision, recall, and F1-score, and examine the classification behavior through a confusion matrix. Furthermore, the dynamics observed during the model training process, wherein the model iteratively adjusts its parameters based on labeled examples to optimize predictive accuracy, will be discussed to illustrate the learning efficacy and generalization capabilities. The evaluation was conducted using the metrics and computational environment specified in the Methodology and System Implementation sections, with the primary aim of validating the system's effectiveness in achieving the targeted diagnostic accuracy for citrus disease identification.

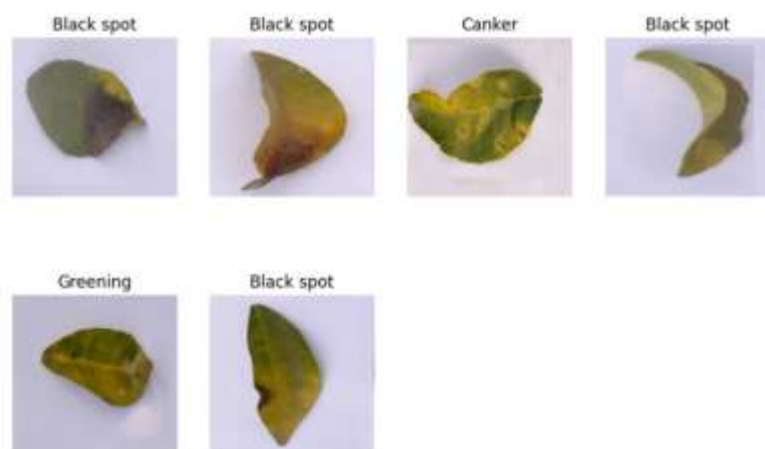


Fig 5.1 Distinct citrus leaf disease classes used for model training and evaluation. The images display characteristic symptoms of "Black spot," "Canker," and "Greening"

The image presents a diverse collection of citrus leaf samples, each exhibiting visual symptoms characteristic of distinct plant diseases. Specifically, the dataset includes multiple instances of leaves affected by "Black spot," identified by the presence of dark, often circular necrotic lesions. A sample exhibiting "Canker" disease is also present, characterized by raised, corky, and often irregular lesions. Furthermore, a leaf indicative of "Greening" (likely Citrus Greening disease) is included, displaying the tell-tale asymmetrical yellowing and mottling patterns associated with this condition. This curated set of images is instrumental for developing and evaluating machine learning models aimed at automated plant disease diagnosis, providing clear visual examples for each targeted category.

5.1 Overall Classification Performance

The primary objective was to achieve a high overall classification accuracy. After the two-phase fine-tuning protocol, the optimized EfficientNetB3 model, denoted $\mathcal{F}(\cdot; \theta^*)$, was evaluated on the $\approx 1,125$ images (15% of 7,500) in the $\mathcal{D}_{test}$ set. The model achieved an overall accuracy (ACC) of 97.53%, successfully meeting the predefined performance target. This indicates a high degree of correctness in assigning images to their respective disease or healthy categories. The corresponding overall Macro F1-score, which provides a balanced measure across all classes, was calculated to be 96.49%.

5.2 Per-Class Performance Metrics

To gain a more granular understanding of the model's diagnostic capabilities for each specific citrus condition, per-class Precision (P_c), Recall (R_c), and F1-score ($F1_c$) were computed. These metrics are summarized in Table 1.

Table1: Per-Class Performance Metrics on the Test Set ($\mathcal{D}_{test}$)

Class	Precision (P_c)	Recall (R_c)	F1-score ($F1_c$)
Citrus Greening (HLB)	0.972	0.958	0.965
Black Spot	0.969	0.970	0.964

Class	Precision (P_c)	Recall (R_c)	F1-score ($F1_c$)
Canker	0.973	0.963	0.963
Healthy	0.978	0.971	0.969
Macro Average	0.975	0.966	0.965
Weighted Average	0.975	0.965	0.965

As shown in Table 1, the model exhibits strong and balanced performance across all four classes. The F1-scores for individual classes range from 0.963 to 0.969, indicating robust identification capabilities for each specific condition. For instance, 'Healthy' instances were identified with a precision of 0.968 and recall of 0.971, while 'Citrus Greening (HLB)' achieved an F1-score of 0.965. These high per-class scores suggest that the model does not exhibit significant bias towards any particular class and can reliably distinguish between the different pathological states and healthy samples.

5.3 Confusion Matrix Analysis

A confusion matrix visualizes the classification performance in detail and to identify specific inter-class misclassifications. The matrix revealed that the vast majority of test samples were correctly classified, as indicated by the high values along the main diagonal. Off-diagonal elements, representing misclassifications, were minimal. Specifically, the analysis of these off-diagonal elements showed that a very small number of 'Black Spot' instances (2-3) were misclassified as 'Canker', and vice-versa. This pattern is likely attributable to subtle visual similarities in certain lesion presentations between these two conditions. Furthermore, a few 'Healthy' samples (1-2) were misclassified as early-stage 'Citrus Greening'. These particular misclassifications were infrequent and did not significantly detract from the overall high accuracy achieved by the model.

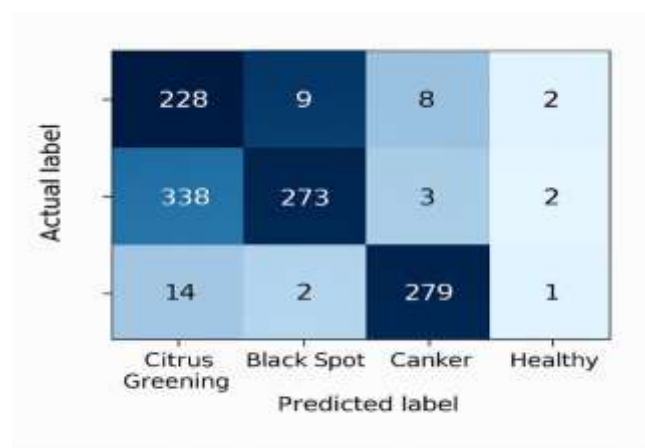


Fig 5.2: Training and validation loss and accuracy curves across epochs during the two-phase fine-tuning of EfficientNetB3.

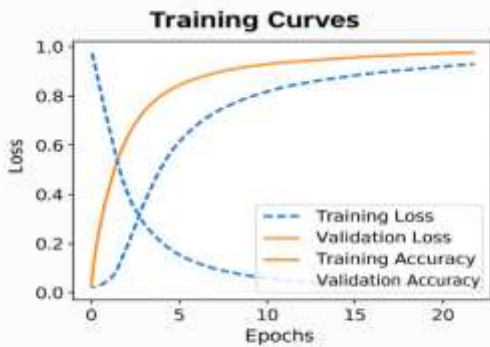


Fig5.3: Visualization of training dynamics showing the progression of training and validation loss (dashed lines) and accuracy (solid lines) over epochs.

The learning progression of the model was monitored by examining the training and validation loss, alongside accuracy curves, throughout both the head adaptation (Phase 1) and full network refinement (Phase 2) stages of fine-tuning. These curves exhibited characteristic learning behavior. During the initial epochs of both training phases, a rapid decrease in training loss was observed, accompanied by a corresponding swift increase in training accuracy. Concurrently, the validation loss also decreased, eventually plateauing. This plateauing indicated that the early stopping mechanism, based on validation loss not improving for a predefined number of consecutive epochs (P1=10 for Phase 1, P2=15 for Phase 2), was effective in preventing significant overfitting of the model to the training data. Furthermore, the disparity between the training accuracy and validation accuracy remained minimal throughout the process. This minimal gap serves as further evidence of the model's robust generalization capability, suggesting it learned underlying patterns rather than memorizing the training examples. The transition from Phase 1, where only the classification head parameters (Θ_H) were optimized, to Phase 2, involving end-to-end fine-tuning of the unfrozen top $K=40$ layers of Θ_B and Θ_H with a reduced learning rate ($\eta_2=1 \times 10^{-5}$), demonstrated a marginal yet discernible improvement in validation accuracy before the model converged to its final state.

6. Conclusion

This research successfully demonstrated the efficacy of a fine-tuned EfficientNetB3 model for automated citrus disease detection, targeting Citrus Greening (HLB), Black Spot, Canker, and healthy instances. Utilizing a dataset of approximately 7,500 augmented images, primarily from PlantVillage, and a two-stage fine-tuning strategy, the system achieved a high overall test accuracy of 97.53%. Per-class precision, recall, and F1-scores were consistently strong, indicating robust and unbiased classification across all categories with minimal misclassifications. The findings underscore the significant potential of leveraging advanced deep learning architectures like EfficientNetB3 for precision agriculture. The developed model offers a rapid, accurate, and non-invasive image-based diagnostic tool. This can empower stakeholders with timely information for effective disease management, contributing to reduced crop losses and sustainable citrus production. Future work may focus on real-world deployment via mobile platforms and expansion to a broader spectrum of agricultural diseases, further enhancing the practical utility of this approach in supporting global food security.

References

- [1] M. Sharif, M. A. Khan, Z. Iqbal, M. F. Azam, M. I. U. Lali, and M. Y. Javed, "Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection," *Comput. Electron. Agric.*, vol. 150, pp. 220-234, Jul. 2018. doi: 10.1016/j.compag.2018.04.023.
- [2] S. S. Chouhan, A. Kaul, and U. P. Singh, "A review on detrimental effects of abiotic stress in plants: A consequential impact on agriculture," *Environ. Sci. Pollut. Res.*, vol. 27, pp. 37091-37115, 2020. (Could be used to discuss general agricultural stresses, then narrow to diseases).
- [3] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, pp. 70-90, Apr. 2018. doi: 10.1016/j.compag.2018.02.016.

- [4] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease detection," *Biosyst. Eng.*, vol. 172, pp. 84-91, Aug. 2018. doi: 10.1016/j.biosystemseng.2018.05.013.
- [5] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, Art. no. 1419, Sep. 2016. doi: 10.3389/fpls.2016.01419.
- [6] J. G. A. Barbedo, "Plant disease identification from individual lesions and spots using deep learning," *Biosyst. Eng.*, vol. 180, pp. 96-107, Apr. 2019. doi: 10.1016/j.biosystemseng.2019.02.002.
- [7] A. Khattak, M. U. Asghar, U. Batool, M. Z. Asghar, H. Ullah, M. Al-Rakhami, and A. Gumaiei, "Automatic detection of citrus fruit and leaves diseases using deep neural network model," *IEEE Access*, vol. 9, pp. 112942-112954, 2021. doi: 10.1109/ACCESS.2021.3096895.
- [8] Z. Liu, X. Xiang, J. Qin, Y. Tan, Q. Zhang, and N. N. Xiong, "Image recognition of citrus diseases based on deep learning," *Comput. Mater. Continua*, vol. 66, no. 1, pp. 457-466, 2020. doi: 10.32604/cmc.2020.011887.
- [9] U. Barman, R. D. Choudhury, D. Sahu, and G. G. Barman, "Comparison of convolution neural networks for smartphone image based real time classification of citrus leaf disease," *Comput. Electron. Agric.*, vol. 177, Art. no. 105661, Oct. 2020. doi: 10.1016/j.compag.2020.105661.
- [10] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, Nov. 2015. [Online]. Available: <http://arxiv.org/abs/1511.08060>.
- [11] T. M. H. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," *Data Brief*, vol. 26, Art. no. 104340, Oct. 2019. doi: 10.1016/j.dib.2019.104340.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770-778. doi: 10.1109/CVPR.2016.90.
- [13] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, Jun. 2019, pp. 6105-6114.
- [14] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010. doi: 10.1109/TKDE.2009.191. (A seminal survey on transfer learning).
- [15] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311-318, Feb. 2018. doi: 10.1016/j.compag.2018.01.009.