*Journal of the Brazilian Computer Society, 2026, 32:1,* doi: 10.5753/jbcs.2026.5712


# Reducing Complexity in Leaf Disease Classification Using a Lightweight Convolutional Neural Network

**Roney Nogueira de Sousa** [ **Federal Institute of Education, Science and Technology of Ceará** | ***roney.nogueira.sousa08@aluno.ifce.edu.br*** ]
**Saulo Anderson Freitas de Oliveira** [ **Federal Institute of Education, Science and Technology of Ceará** | ***saulo.oliveira@ifce.edu.br*** ]
**Pedro Pedrosa Reboucas Filho** [ **Federal Institute of Education, Science and Technology of Ceará** | ***pedrosarf@ifce.edu.br*** ]

*Federal Institute of Education, Science and Technology of Ceará, Av. 13 de Maio, 2081, Benfica, Fortaleza, CE, 60040-215, Brazil.*

**Received:** 20 March 2025 • **Accepted:** 21 August 2026 • **Published:** 28 August 2026

**Abstract.** Agricultural production faces significant annual losses due to plant diseases, with economic impacts exceeding 40 million dollars and contributing to acute hunger affecting over 281.6 million people in 2023. The timely and accurate identification of plant diseases through leaf image analysis is crucial to mitigate these losses and ensure global food security. This study proposes a lightweight Convolutional Neural Network model, inspired by the MobileNet architecture, designed to classify various plant leaf diseases efficiently. Leveraging a publicly available dataset, this research focuses on developing a model that balances high performance with computational efficiency, making it suitable for real-world applications in resource-constrained environments. The proposed model, named LDPNet, achieved an outstanding accuracy of 99.62%, alongside precision, recall, F1-Score, and AUC metrics of 99.16%, 99.08%, 99.11%, and 99.99%, respectively. A comprehensive comparative analysis was conducted against MobileNetV2 and a reference model from previous research, highlighting the superior performance of LDPNet in terms of both accuracy and efficiency. The results demonstrate that the proposed architecture not only maintains high classification performance but also significantly reduces the number of parameters, making it a practical and scalable solution for plant disease identification. This study contributes to the growing field of agricultural technology by providing a robust, lightweight, and efficient tool for early disease detection, with the potential to enhance crop management and reduce economic losses in agriculture.

**Keywords:** Plant disease identification, Lightweight CNN, LDPNet, Deep learning, Agriculture.

## 1 Introduction

Each year, up to 40% of the world's agricultural output is compromised by plant pests and diseases, leading to massive global economic damages exceeding 220 billion USD annually Food and Agriculture Organization of the United Nations (FAO) [2024]; Gai and Wang [2024]. This profound biological vulnerability severely threatens global food security, compounding a critical crisis where between 638 and 720 million individuals faced hunger globally in 2024 Food and Agriculture Organization of the United Nations (FAO) [2024]; WHO [2024]. The macroeconomic impact is especially severe in primary agricultural hubs such as Brazil, which plays a central role in the global food supply but faces continuous threats from tropical pathogens Angelotti *et al*. [2024]. Within the Brazilian context, specific endemic outbreaks generate catastrophic financial losses; for instance, the recent epidemic of corn stunt diseases alone caused an estimated annual direct loss of 6.5 billion USD, destroying nearly 32 million tons of grain per year de Oliveira *et al*. [2026].

Farmers routinely allocate substantial resources towards disease prevention and control; however, limited access to effective technical guidance often results in insufficient management and poor outcomes Huber and Jones [2013]; Husson *et al*. [2021]. Foliar diseases, in particular, pose severe threats due to their rapid transmission and potential to cause substantial harm to crops quickly. If not swiftly identified and precisely managed, these diseases can result in complete agricultural losses. Consequently, leaf diseases pose significant risks to global agriculture, substantially decreasing crop yields, deteriorating fruit quality, and diminishing nutritional value, collectively leading to reduced income for agricultural producers MOININA *et al*. [2019].

The identification of plant diseases through leaf image analysis has become a crucial area of research in agriculture, as early detection is essential to mitigating production losses. However, traditional disease identification methods often entail high labor costs, a lack of real-time monitoring, and the risk of misdiagnosis by non-specialists, all of which complicate disease management and contribute to reduced yields and food security concerns Khirade and Patil [2015]; Ferentinos [2018]. Nonetheless, findings from Bharate and Shirdhonkar [2017] indicate that most leaf diseases exhibit distinct visual symptoms, underscoring the necessity for learning models capable of accurately recognizing these patterns.

To facilitate the identification of visible symptoms, Convolutional Neural Networks (CNNs) can be employed. These specialized neural networks are designed to process structured data, such as images represented in a two-dimensional pixel grid or sequential data captured in one-dimensional samples Goodfellow *et al*. [2016]. The integration of CNNs has led to substantial advancements in multiple fields, including agriculture, enhancing the efficiency and accuracy of disease detection in plants.

The classification of plant diseases using CNNs has been extensively explored in the literature, often targeting specific crops. For instance, Funck [2019] proposed a mobile application for identifying soybean Asian rust, achieving an accuracy of 84.61%. Similarly, Lu *et al*. [2017] employed a modified AlexNet architecture Krizhevsky *et al*. [2017] for detecting rice diseases, attaining a recognition rate of 95.4%. Additionally, an Inception network-based model Szegedy *et al*. [2015] was utilized by Li *et al*. [2020] to classify diseases in eggplants, yielding an accuracy of 95.09%.

Artificial intelligence techniques have demonstrated their potential in addressing complex agricultural challenges and enhancing productivity. This study specifically focuses on developing a lightweight CNN model, inspired by the MobileNet architecture Howard *et al*. [2017], to classify various plant leaf diseases using image data. The research draws inspiration from *"Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network"* by Geetharamani and Pandian [2019], leveraging the same publicly available dataset entitled *"Data for: Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network"*[1]. Since this dataset had been augmented by the original author, it enables direct comparisons of performance across different models.

The proposed CNN architecture achieved an accuracy of 99.62%. Other performance metrics include a precision of 99.16%, recall of 99.08%, F1-score of 99.11%, and an area under the ROC curve (AUC) of 99.99%. These findings highlight the model's capability to provide consistent and reliable classifications, reinforcing its effectiveness and robustness in plant disease detection.

A preliminary version of this research was presented in Sousa *et al*. [2024], where a lightweight CNN model, inspired by MobileNet, was introduced for plant disease classification. While the conference version demonstrated promising results, this extended version includes several key improvements and contributions:

- **Per-Class Accuracy Analysis:** Unlike the conference version, which focused on overall model performance, this study includes an in-depth per-class accuracy evaluation. A detailed comparison table illustrates how different disease categories are classified and how the model performs relative to the reference work.
- **Impact of Batch Size on Model Behavior:** This version includes a systematic analysis of how batch size affects model performance. By varying the batch size, we explore its influence on accuracy, training stability, and computational efficiency.
- **Comparison with the Reference Model:** A direct comparison is made with the reference model, highlighting improvements in accuracy per class and overall classification reliability. This section provides insights into how the proposed LDPNet model enhances classification quality while maintaining efficiency.
- **Comparison with Related Works:** This version extends the analysis by including a section that compares LDPNet with other recent studies in the field. The comparison focuses on performance metrics, dataset usage, and computational efficiency, demonstrating how our approach stands out among state-of-the-art models.
- **Optimizer Change:** In contrast to the conference version, which employed the RMSProp optimizer, this study adopts the Adam optimizer. This change enhances model convergence speed and stability, reducing sensitivity to hyperparameter tuning and improving overall classification performance.
- **More Detailed Statistical Results:** This extended version provides a more in-depth statistical analysis of the results, incorporating additional performance metrics such as standard deviation, confidence intervals, and detailed error distributions. This allows for a deeper understanding of model reliability and classification consistency across different classes.

These enhancements justify the submission of this extended work, as it provides a more comprehensive and practically applicable solution for plant disease identification.

## 1.1 Document Organization

This work is organized into four main sections, each addressing a crucial aspect of the research to provide a comprehensive understanding of the proposed methodology, results, and contributions. The document is structured as follows:

**Section 3** provides a review of related works, presenting the state of the art in plant disease identification using deep learning techniques.

**Section 4** outlines the methodology adopted in this study. It describes the dataset, detailing its composition, preprocessing steps, and augmentation techniques. Furthermore, it explains the network training process, covering hyperparameter tuning, regularization strategies, and the architecture of the proposed lightweight CNN. The section concludes with a discussion of the experimental setup and evaluation framework.

**Section 5** discusses the experimental results obtained with the proposed model. The evaluation metrics—accuracy, precision, recall, F1-Score, and AUC—are analyzed in detail. Additionally, the performance of the proposed model is compared with MobileNetV2 and the reference model from Geetharamani and Pandian [2019]. A comparative analysis with state-of-the-art approaches from the literature is also included, along with an evaluation of the impact of data augmentation and batch size on model performance.

**Section 6** presents the concluding remarks, summarizing the key findings and contributions of this work. It also discusses the study's limitations and suggests potential future research directions, including the exploration of advanced architectures.

[1] `https://data.mendeley.com/datasets/tywbtsjrjv/1`

## 2 Objectives

This study aims to develop a lightweight and efficient CNN model for plant leaf disease identification, contributing to the reduction of agricultural losses and increased productivity. To achieve this overarching goal, the following specific objectives have been defined:

1. **Design a lightweight CNN architecture**: Develop a convolutional neural network inspired by the MobileNet model, optimizing the number of parameters while ensuring high accuracy and computational efficiency.
2. **Assess the model's performance**: Compare the proposed architecture against state-of-the-art models, including MobileNetV2 and the reference model from Geetharamani and Pandian [2019], using key evaluation metrics such as accuracy, precision, recall, F1-Score, and AUC.
3. **Analyze the impact of data augmentation**: Investigate how different data augmentation techniques influence model performance under both balanced and imbalanced class distributions.
4. **Optimize the training process**: Apply regularization techniques, including early stopping and hyperparameter tuning, to mitigate overfitting and enhance the model's generalization capability.
5. **Develop a practical and deployable model**: Ensure the model is suitable for deployment on resource-constrained devices, such as smartphones and embedded systems, enabling real-world application by farmers and agricultural professionals.

## 3 Related Works

A study by Syamsuri and Kusuma [2019] explored the optimization of mobile-based plant disease detection using CNN models. The PlantVillage dataset, extended to include coffee leaves, was used to evaluate three CNN models: MobileNet, Mobile Nasnet (MNasNet), and InceptionV3. The results indicated that InceptionV3 achieved the highest accuracy of 95.79%. In terms of performance metrics, MobileNet achieved approximately 95% precision, 94% recall, and 95% F1-score. MNasNet had approximately 97% precision, 96% recall, and 96% F1-score. InceptionV3 showed approximately 96% precision, 96% recall, and 96% F1-score.

In KC *et al*. [2019], depthwise separable convolution architectures were proposed for plant disease detection using a subset of the PlantVillage dataset with 82,161 images across 55 classes. The Reduced MobileNet achieved an accuracy of 98.34%.

In the work of Gajjar *et al*. [2022], a hybrid system for real-time plant disease detection was proposed, combining leaf detection and disease classification. Using the PlantVillage dataset, a Single Shot Multibox Detector (SSD) model was trained for detecting leaves, which were then passed through a CNN for disease classification. The system, implemented on an embedded platform, achieved an accuracy of 96.88%, demonstrating its effectiveness in identifying multiple diseases in real-time from leaf images.

In the work of Falaschetti *et al*. [2022], an image detector using a resource-constrained Convolutional Neural Network (CNN) was proposed for real-time plant disease classification. The CNN was trained on the ESCA-dataset and the PlantVillage-augmented dataset, and implemented on the low-cost OpenMV Cam H7 Plus platform. The system performed real-time image acquisition and classification, displaying results on an LCD screen. Experimental results showed an accuracy of 98.10% and 95.24% on the ESCA and PlantVillage-augmented datasets, respectively.

Leveraging advanced deep learning techniques, Pandian *et al*. [2022] introduced a DCNN-based solution for identifying 26 distinct plant diseases from leaf images. To expand the dataset from 55,448 to 234,000 images, the authors employed innovative augmentation methods, including Neural Style Transfer, DCGAN, Principal Component Analysis, and Color and Position Augmentation, achieving a balanced dataset with 6,000 images per class. The Conv-5 DCNN model, trained on this enhanced dataset, underwent hyperparameter optimization through random search. The proposed model achieved 98.41% accuracy, with precision, recall, and F1-scores of approximately 94%, 100%, and 97%, respectively.

In de Sousa and de Sousa [2023], a Convolutional Neural Network (CNN) architecture was proposed for plant disease classification using the PlantVillage dataset. The model was trained and tested using 54,305 images across 39 disease classes. To address overfitting, data augmentation techniques were applied during training. Experimental results showed that the CNN model achieved an accuracy of 99.55%, precision of 94.25%, AUC of 99.81%, and F1-Score of 90.86%.

Harnessing the complementary strengths of deep learning architectures, Hashemifar and Zakeri [2024] introduced a hybrid model that combined the global context modeling of Pyramid Vision Transformer (PVT) with the feature extraction prowess of VGG-16 for plant disease classification. Evaluated on the PlantVillage dataset, the model demonstrated remarkable performance, achieving an accuracy of 98.51%, a precision of 97%, a recall of 98%, and an F1-Score of 97%.

## 4 Methodology

This study aimed to train and evaluate a new CNN architecture for classifying leaf diseases in images. The network's capability to identify and categorize diseases was assessed through performance metrics analysis. All procedures adopted in this study are detailed in the following subsections.

### 4.1 Dataset

The dataset used in this research is the same as in the study *"Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network"*, publicly available under the title *"Data for: Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network"*[2]. This dataset had already undergone augmentation by its original authors.

Initially, the dataset contained 54,305 images representing 13 different plant species, categorized into 38 disease classes. Each class was labeled as either healthy or infected based on

[2] `https://data.mendeley.com/datasets/tywbtsjrjv/1`

**Table 1.** Summary of Metrics Achieved by Related Works

| Author | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | AUC (%) |
|---|---|---|---|---|---|
| Syamsuri and Kusuma [2019] | 95.79 | 96.00 | 96.00 | 96.00 | - |
| KC *et al*. [2019] | 98.34 | - | - | - | - |
| Gajjar *et al*. [2022] | 96.88 | - | - | - | - |
| Falaschetti *et al*. [2022] | 95.24 | - | - | - | - |
| Pandian *et al*. [2022] | 98.41 | 94.00 | 100.00 | 97.00 | - |
| de Sousa and de Sousa [2023] | 99.55 | 94.25 | - | 90.86 | 99.81 |
| Hashemifar and Zakeri [2024] | 98.51 | 97.00 | 98.00 | 97.00 | - |

**Table 2.** Datasets and Technique used by related works

| Author | Model | Dataset | Plants |
|---|---|---|---|
| Syamsuri and Kusuma [2019] | InceptionV3 | PlantVillage | Apple, Blueberry, Cherry, Coffee, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato; |
| KC *et al*. [2019] | MobileNet | PlantVillage | Apple, Banana, Blueberry, Cabbage, Cassava, Cherry, Corn, Cucumber Grape, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato; |
| Gajjar *et al*. [2022] | Custom | PlantVillage | Apple, Corn, Potato, Tomato |
| Falaschetti *et al*. [2022] | Custom | PlantVillage and ESCA | Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato; |
| Pandian *et al*. [2022] | Custom | PlantVillage | Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato; |
| de Sousa and de Sousa [2023] | Custom VGG19 | PlantVillage | Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato; |
| Hashemifar and Zakeri [2024] | Vision Transformers + VGG19 | PlantVillage | Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato; |

the disease type. Additionally, the dataset included a separate background class with 1,143 images. Figure 1 presents sample images from different dataset classes.

Table 3 provides a detailed distribution of images per class before and after data augmentation. The augmentation process significantly increased the number of samples in underrepresented classes, ensuring a more balanced dataset and minimizing potential biases during model training. For instance, the “Apple with cedar apple rust” class expanded from 275 to 1,000 images, with similar increases observed in other initially low-sample classes.

To enhance model robustness and reduce overfitting risks, various data augmentation techniques were applied. These transformations included image flipping, gamma correction, noise injection, PCA color augmentation, rotation, and scaling. As a result, the dataset expanded to a total of 61,486 images, providing a more balanced and diverse training set.

## 4.2 Classification Models

Two CNN models were trained to classify the images in the dataset: the MobileNetV2 CNN Sandler *et al*. [2018] and the proposed CNN architecture developed in this study. For the MobileNetV2 model, a pre-trained version from the Torchvision library was utilized, with weights previously trained on the ImageNet dataset[3].

This approach, known as Transfer Learning, allows knowledge acquired in one domain to be transferred to another by leveraging the structure and parameters of a pre-trained network. Transfer Learning helps mitigate overfitting, as the models have already been regularized and generalized during their initial training Zhuang *et al*. [2020]; Tan *et al*. [2018]. Additionally, it reduces computational costs and accelerates training while enhancing classifier performance. The training was conducted on a machine with the following specifications:

- AMD Ryzen 5 5600X processor (4.6 GHz, 6 physical cores, 12 threads);
- 32 GB of RAM;
- 1.5 TB storage capacity;
- NVIDIA GeForce RTX 2060 Super GPU with 8 GB of dedicated memory.

Both CNNs were trained for up to 3,000 epochs. However, to prevent overfitting, an early stopping mechanism was implemented, monitoring the F1-Score metric. If no improvement was observed after 50 consecutive epochs, training was halted, and the model was reverted to the weights corresponding to the best epoch.

[3] `https://image-net.org/`

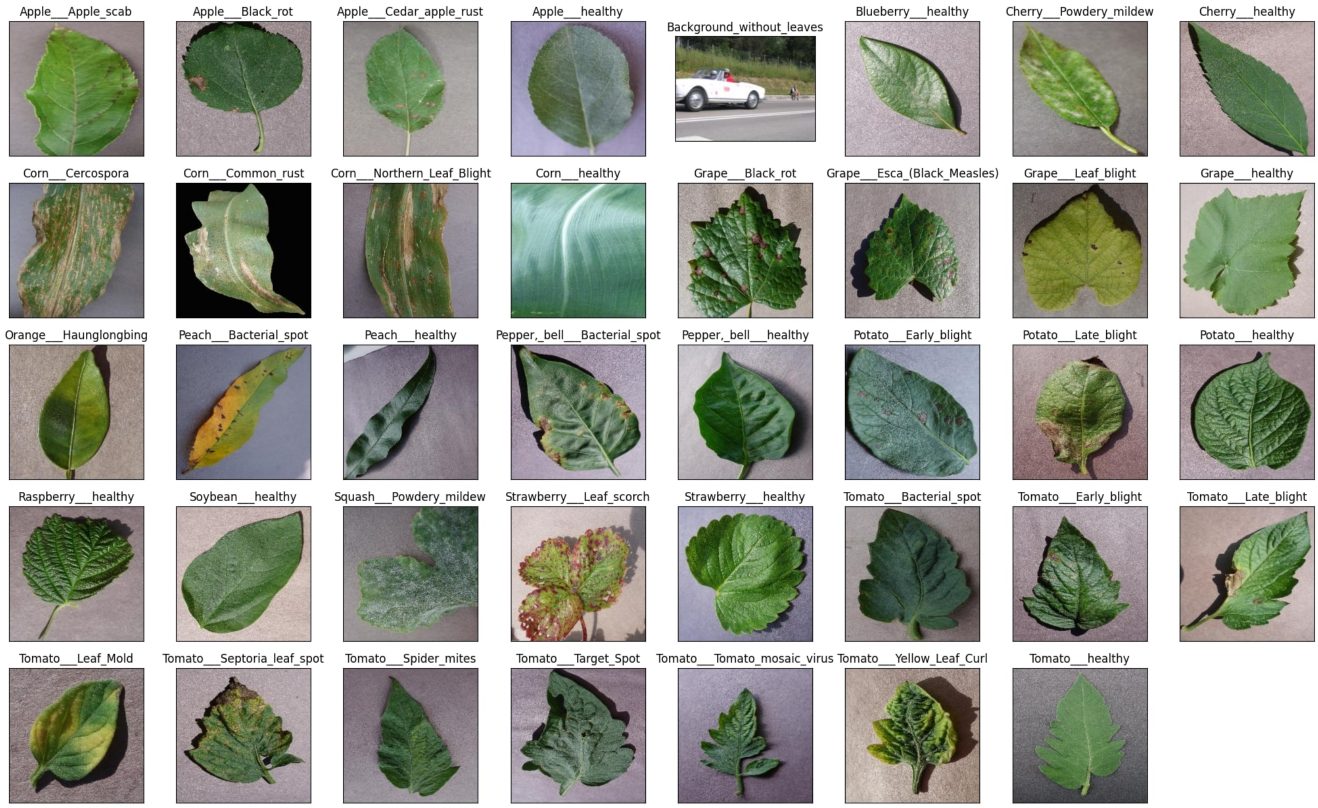


**Figure 1.** Sample Images from the Dataset

During training, the entire dataset was processed in each epoch. After every epoch, model performance was evaluated using the validation set, and the results were recorded. Once training was completed, both models were tested on the test set for final performance assessment.

Following the methodology of the reference study, experiments were conducted under two conditions: (i) **Scenario 1** – training without data augmentation, and (ii) **Scenario 2** – training with data augmentation. This comparative analysis aimed to assess the influence of data augmentation on model performance and determine which scenario led to better results.

## 4.3 *Lightweight Disease Prediction Network*

The proposed CNN architecture, named *Lightweight Disease Prediction Network* (LDPNet), was designed based on the principles of MobileNet, which employs depthwise separable convolutions. These convolutions significantly reduce the number of parameters and computational operations while maintaining strong performance. The network structure integrates convolutional and pooling layers, enabling multi-level feature extraction similar to MobileNet but with a more compact architecture. Table 4 presents a comparison of the parameter count among the models used in this study.

As shown in Table 4, LDPNet has the lowest number of parameters among the evaluated architectures. This demonstrates the effectiveness of the proposed model in reducing complexity while maintaining competitive performance. A lower parameter count translates into reduced computational resource requirements, making the model more efficient and suitable for deployment on resource-constrained devices.

The structure of LDPNet is illustrated in Figure 2. The architecture begins with an initial convolutional layer, followed by batch normalization and a ReLU activation function. Subsequently, a series of MobileNet blocks are employed. Each MobileNet block consists of depthwise separable convolutions and pointwise convolutions, enabling the network to learn spatial filters independently for each input channel. Each pointwise convolution is followed by batch normalization and a ReLU activation function.

At the final stage of the model, an adaptive pooling layer reduces the feature maps to a 1×1 feature vector, minimizing dimensionality while preserving essential information. This vector is then processed through a dense layer with a softmax activation function to predict class probabilities across the 39 categories. The proposed architecture demonstrates that a compact model with fewer parameters can still achieve robust and efficient performance, balancing computational efficiency and classification accuracy.

For a detailed breakdown of the architecture, Table 5 outlines the configuration of LDPNet's layers, specifying layer type, output dimensions, and parameter counts.

The proposed LDPNet model introduces a few advantages over traditional CNN architectures:

- **Reduced Computational Complexity:** By incorporating depthwise separable convolutions, LDPNet achieves a lower parameter count than MobileNetV2 and other state-of-the-art architectures, improving efficiency without sacrificing performance.
- **Improved Inference Speed:** A compact architecture allows for faster processing times, making the model suitable for real-time applications, such as mobile-based

**Table 3.** Distribution of images per class in the leaf disease dataset.

| Class Name | Before Data Augmentation | After Data Augmentation |
|---|---|---|
| Apple with scab | 630 | 1000 |
| Apple with black rot | 621 | 1000 |
| Apple with cedar apple rust | 275 | 1000 |
| Healthy apple | 1645 | 1645 |
| Blueberry (healthy) | 1502 | 1502 |
| Cherry with powdery mildew | 1052 | 1052 |
| Healthy cherry | 854 | 1000 |
| Corn with grey leaf spot | 513 | 1000 |
| Corn with common rust | 1192 | 1192 |
| Corn with northern leaf blight | 985 | 1000 |
| Healthy corn | 1162 | 1162 |
| Grape with black rot | 1180 | 1180 |
| Grape with black measles | 1383 | 1383 |
| Grape with leaf blight | 1076 | 1076 |
| Healthy grape | 423 | 1000 |
| Orange with Huanglongbing | 5507 | 5507 |
| Peach with bacterial spot | 2297 | 2297 |
| Healthy peach | 360 | 1000 |
| Pepper with bacterial spot | 997 | 1000 |
| Healthy pepper | 1478 | 1478 |
| Potato with early blight | 1000 | 1000 |
| Healthy potato | 152 | 1000 |
| Potato with late blight | 1000 | 1000 |
| Healthy raspberry | 371 | 1000 |
| Healthy soybean | 5090 | 5090 |
| Squash with powdery mildew | 1835 | 1835 |
| Healthy strawberry | 456 | 1000 |
| Strawberry with leaf scorch | 1109 | 1109 |
| Tomato with bacterial spot | 2127 | 2127 |
| Tomato with early blight | 1000 | 1000 |
| Healthy tomato | 1591 | 1591 |
| Tomato with late blight | 1909 | 1909 |
| Tomato with leaf mold | 952 | 1000 |
| Tomato with septoria leaf spot | 1771 | 1771 |
| Tomato with two-spotted spider mite | 1676 | 1676 |
| Tomato with target spot | 1404 | 1404 |
| Tomato with mosaic virus | 373 | 1000 |
| Tomato with yellow leaf curl virus | 5357 | 5357 |
| Background (without leaf) | 1143 | 1143 |
| Total Images | 55,448 | 61,486 |

**Table 4.** Comparison of Parameter Counts in Different Architectures

| **Model** | **Total Parameters** |
|---|---|
| LDPNet | 1,817,063 |
| MobileNetV2 | 2,273,831 |
| Reference Model from Literature | 2,672,735 |

plant disease detection.

- **Optimized for Low-Resource Devices:** With its lightweight design, LDPNet is an excellent candidate for deployment on resource-constrained hardware, such as smartphones and edge devices used in agriculture.

**Table 5.** Layer Configuration of the Proposed Convolutional Neural Network

| **Layer Type** | **Output Shape** | **Parameters** |
|---|---|---|
| Convolutional | 64×64×32 | 896 |
| Batch Normalization | 64×64×32 | 128 |
| MobileNet Block 1 | 64×64×64 | 2,432 |
| MobileNet Block 2 | 32×32×128 | 9,152 |
| MobileNet Block 3 | 16×16×256 | 35,328 |
| MobileNet Block 4 | 8×8×512 | 138,576 |
| MobileNet Block 5 | 8×8×1024 | 1,055,328 |
| Global Average Pooling | 1×1×1024 | 0 |
| Dense (Softmax Output) | 39 | 39,975 |

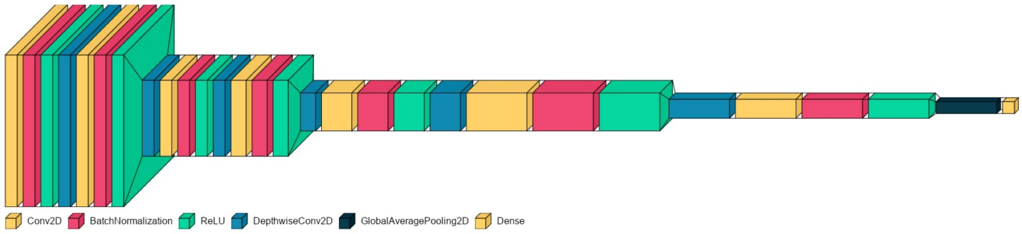


**Figure 2.** Proposed Convolutional Neural Network Architecture

### 4.4 Training Process

The training process of LDPNet was designed to ensure efficient convergence and optimal performance. The Adam optimizer Kingma [2014] was used to train the model due to its effectiveness in deep learning tasks. Adam, which stands for *Adaptive Moment Estimation*, combines the benefits of two other popular optimization algorithms: AdaGrad Duchi *et al*. [2011] and RMSProp [4]. It adapts the learning rate for each parameter individually, using estimates of the first and second moments of the gradients. This adaptive approach allows for faster convergence and better handling of sparse gradients, making it particularly suitable for training deep neural networks.

Adam operates by updating the model parameters as follows:

1. Computes the gradient of the loss function with respect to the parameters:
$$g_t = \nabla_\theta J(\theta_t),$$
where $g_t$ is the gradient at step $t$, $J(\theta_t)$ is the loss function, and $\theta_t$ are the model parameters at step $t$.
2. Updates the estimates of the first moment (mean) and the second moment (uncentered variance) of the gradients:
$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t,$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2,$$
where $m_t$ and $v_t$ are the estimates of the first and second moments, respectively, and $\beta_1$ and $\beta_2$ are the exponential decay rates.
3. Corrects the bias in the first and second moment estimates, as they are initialized at zero:
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t},$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}.$$
4. Updates the model parameters:
$$\theta_{t+1} = \theta_t - \alpha\frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon},$$
where $\alpha$ is the learning rate and $\epsilon$ is a small constant to avoid division by zero.

For this study, the Adam optimizer was configured with the following hyperparameters:

- Learning rate ($\alpha$): $1 \times 10^{-4}$
- Beta1 ($\beta_1$): $0.9$
- Beta2 ($\beta_2$): $0.999$
- Epsilon ($\epsilon$): $1 \times 10^{-8}$

These hyperparameters were chosen empirically and based on common practices in the deep learning literature. The learning rate, in particular, was selected to balance convergence speed and stability.

## 5 Results and Discussions

This section discusses the results obtained from the methodology presented in Section 4 for the classification of leaf diseases based on images.

### 5.1 Influence of Batch Size on Model Performance

The Table 6 presented the accuracy of the proposed model as a function of the batch size, evaluating two distinct scenarios: without data augmentation (Scenario 1) and with data augmentation (Scenario 2). These results highlighted the model's sensitivity to the batch size, which was a critical parameter for optimizing performance during training.

In both analyzed scenarios, the model achieved higher accuracy with a batch size of 16, emphasizing the effectiveness of smaller batches in capturing patterns within the data. In Scenario 1, without data augmentation, accuracy dropped significantly with larger batches, reaching 91.80% with a batch size of 256, due to reduced variability in weight updates. In

[4]`https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`

Scenario 2, with data augmentation, accuracy was consistently higher across all configurations, peaking at 99.62% with a batch size of 16. The performance decline with larger batches was less pronounced, demonstrating that data augmentation improved model generalization and mitigated the negative impacts of larger batch sizes.

An interesting aspect to highlight was the relative performance difference between the two scenarios. With a batch size of 16, the accuracy in Scenario 2 was 0.36 percentage points higher than in Scenario 1, indicating that data augmentation enhanced the utility of smaller batch sizes. On the other hand, with a batch size of 256, this difference increased to 0.82 percentage points, suggesting that data augmentation had an even greater impact in configurations with larger batches, potentially compensating for the reduced frequency of weight updates.

The results also underscored the challenge of balancing batch size with computational efficiency. While smaller batches offered better accuracy performance, they increased training time due to the higher frequency of weight updates. Conversely, larger batches, despite reducing accuracy, may have been more advantageous in terms of computational efficiency in resource-constrained scenarios.

## 5.2 Results Scenario 1

In this subsection, we presented the performance of CNN architectures without the use of data augmentation, analyzing how data availability influenced the learning process of the models. The evaluation metrics, obtained from the test set, were shown in Table 7.

When comparing the performance of the two models in Scenario 1, it was observed that the proposed architecture outperformed MobileNetV2 in all key metrics, except for AUC, where both models performed well.

The results table indicated that LDPNet achieved an accuracy of 99.62%, while MobileNetV2 achieved 95.18%, a difference of 4 percentage points, suggesting that the proposed model offered more accurate overall classification. In terms of precision, LDPNet reached 98.93%, demonstrating a lower false-positive rate compared to MobileNetV2, which achieved 94.36%, indicating higher effectiveness in avoiding unnecessary treatment for healthy plants. The recall of the proposed model was 98.47%, superior to MobileNetV2's 93.28%, showing greater ability to correctly identify diseased plants. For the F1-Score, LDPNet maintained a more effective balance between precision and recall, with 98.69%, outperforming MobileNetV2, which scored 93.81%. Finally, the AUC metric for LDPNet was 99.98%, while MobileNetV2 achieved 99.09%, indicating that although both models demonstrated good discriminative ability, the proposed model achieved superior performance.

The results obtained in Scenario 1 revealed that LDPNet, although simpler and less complex than MobileNetV2, managed to achieve superior performance. This indicated that simplifying the model did not compromise the quality of predictions, but rather enhanced its ability to handle the specific task of classifying leaf diseases. These results suggested that the reduction in architectural complexity not only preserved performance but also provided a more practical and viable solution for the proposed task. This structural efficiency was the true differentiator of the model, demonstrating that it was not necessary to rely on more complex and heavier architectures to achieve generalization capacity and performance.

## 5.3 Results Scenario 2

This subsection aimed to show the performance of CNN architectures when data augmentation techniques were applied, thus verifying whether greater data availability could positively impact the models' learning process. These evaluation metrics were shown in Table 8.

In Scenario 2, it was observed that the application of data augmentation techniques resulted in improvements across all metrics for both models. LDPNet showed an overall improvement in metrics compared to Scenario 1: accuracy increased to 99.62%, with an increment of 0.36 percentage points; precision rose to 99.16%, an increase of 0.23 percentage points; recall reached 99.08%, with an improvement of 0.61 percentage points; F1-Score increased to 99.11%, improving by 0.42 percentage points; and AUC reached 99.99%, a slight increase of 0.01 percentage points.

For MobileNetV2, data augmentation was also beneficial. Accuracy improved to 98.04%, a gain of 2.86 percentage points; precision increased to 97.45%, with an increment of 3.09 percentage points; recall reached 97.25%, an improvement of 3.97 percentage points; F1-Score increased to 97.34%, improving by 3.53 percentage points; and AUC was 99.96%, an improvement of 0.87 percentage points.

In summary, data augmentation proved to be a significant factor in improving the generalization ability and performance of the models, both for the proposed model and for MobileNetV2. However, LDPNet not only benefited from these techniques but also stood out for its simplicity and superior efficiency. The analysis reinforced that while data augmentation techniques are essential for optimizing model performance in challenging contexts, the combination of a simplified architecture with data augmentation can offer a more practical and efficient solution for the task of classifying leaf diseases.

## 5.4 Comparison with the Reference Article

This subsection compares the results obtained with those presented in the article "Identification of plant leaf diseases using a nine-layer deep convolutional neural network" Geetharamani and Pandian [2019], which inspired this work. The goal is to evaluate the relative performance of the developed convolutional neural network models, highlighting the improvements achieved and the impact of simplifying the architecture.

The architecture from the reference article, composed of multiple convolutional and pooling layers followed by dense layers, represented a classic and relatively complex approach to the classification task, totaling 2,672,735 parameters. In contrast, the model proposed in this work is simpler, reflecting a significant reduction in structural complexity, with a total of 1,817,063 parameters. This reduction of approximately 32% in the number of parameters not only made the model lighter but also demonstrated that competitive performance could be achieved without an excessively complex model.

**Table 6.** Impact of batch size on model accuracy.

| Batch size | Accuracy (%) | |
|---|---|---|
| | Scenario 1 (without augmentation) | Scenario 2 (with augmentation) |
| 16 | 99.26 | **99.62** |
| 32 | 98.64 | 99.04 |
| 64 | 97.08 | 97.81 |
| 128 | 93.12 | 94.27 |
| 256 | 91.80 | 92.62 |

**Table 7.** Performance Comparison in Scenario 1

| Metric | MobileNetV2 | LDPNet |
|---|---|---|
| Accuracy | 95.18% | **99.26**% |
| Precision | 94.36% | **98.93**% |
| Recall | 93.28% | **98.47**% |
| F1-Score | 93.81% | **98.69**% |
| AUC | 99.09% | **99.98**% |

**Table 8.** Test Set Results in Scenario 2

| Metric | MobileNetV2 | LDPNet |
|---|---|---|
| Accuracy | 98.04% | **99.62%** |
| Precision | 97.45% | **99.16%** |
| Recall | 97.25% | **99.08%** |
| F1-Score | 97.34% | **99.11%** |
| AUC | 99.96% | **99.99%** |

A limitation of the reference article was the lack of evaluation on a test set without data augmentation techniques, reporting only the validation accuracy as 91.43%. The absence of tests without data augmentation limited the comprehensive evaluation of the model's generalization ability. Therefore, the comparison here was based solely on the scenario with data augmentation.

Table 9 presents a comparison of the performance metrics between LDPNet and the model from the reference article, considering accuracy, precision, recall, and F1-Score.

**Table 9.** Comparison of Evaluation Metrics with the Reference Article

| Metric | Reference Model | LDPNet |
|---|---|---|
| Accuracy | 96.46% | **99.62%** |
| Precision | 96.47% | **99.16%** |
| Recall | **99.89%** | 99.08% |
| F1-Score | 98.15% | **99.11%** |

The results in Table 9 showed that LDPNet achieved 99.62% accuracy, surpassing the reference model by 3.16 percentage points. This indicated a superior ability of the simplified model to make correct predictions, supporting the hypothesis that simplification could be effective without sacrificing overall precision. With a precision of 99.16%, the proposed model outperformed the reference model by 2.69 percentage points. This suggested that the reduced complexity may have contributed to better efficiency in minimizing false positives, a crucial aspect in the practice of disease diagnosis. The reference model achieved a recall 0.78 percentage points higher than LDPNet. While the reference model showed a higher capacity to identify positive cases, the difference was relatively small and reflected the effectiveness of the simpler architecture in maintaining good overall performance. The proposed model achieved an F1-Score of 99.11%, exceeding the reference model by 0.96 percentage points. This result demonstrated that, despite the reduced complexity, LDPNet was able to effectively balance precision and recall.

In summary, LDPNet outperformed the reference model in accuracy, precision, and F1-Score, while the reference model had slightly higher recall. These results confirmed that simplifying the architecture could offer a practical and efficient solution while maintaining robust performance. The simplified approach not only improved accuracy and precision but also demonstrated that high efficiency could be achieved with less complex models, avoiding the need for overly complicated structures.

Table 10 shows a detailed comparison of testing accuracy for each class in the dataset between the proposed model and the reference model. This analysis highlighted the effectiveness of the proposed model in classifying leaf disease images.

The results revealed that the proposed model outperformed the reference model in most classes. For instance, in categories like "Apple with scab", "Apple with black rot", and "Corn with grey leaf spot" the proposed model achieved significantly higher accuracies, scoring 99.62%, 99.83%, and 99.13%, respectively. Additionally, in more challenging classes such as "Grape with black rot" and "Potato with late blight", the proposed model demonstrated substantial improvements, showcasing its ability to handle complex cases effectively.

Notably, both models achieved an accuracy of 100% in some classes, such as "Blueberry with healthy", "Cherry with healthy", and "Healthy raspberry". This indicates that these categories contained distinct features that both models were able to classify with ease. However, for the class "Healthy apple", the reference model slightly outperformed the proposed model, achieving 97.00% compared to 96.73%. This highlights a specific area where the proposed model could be refined.

Furthermore, the proposed model achieved perfect accuracy (100%) in 12 classes, reflecting its robustness and superior generalization ability across diverse scenarios. This analysis underscores that the proposed model consistently delivered higher performance across various leaf disease categories.

Moreover, the proposed model achieved perfect accuracy (100%) in 12 classes, reflecting its robustness across various scenarios. This analysis showed that the proposed model consistently delivered higher performance and generalization across different categories.

**Table 10.** Comparison of testing accuracy for each class (in percentages).

| S. No. | Class Name | LDPNet (%) | Reference Model (%) |
|---|---|---|---|
| 1 | Apple with scab | **99.62%** | 96.00% |
| 2 | Apple with black rot | **99.83%** | 98.00% |
| 3 | Apple with cedar apple rust | **99.21%** | 98.00% |
| 4 | Healthy apple | 96.73% | **97.00%** |
| 5 | Blueberry with healthy | **100.00%** | **100.00%** |
| 6 | Cherry with powdery mildew | **99.51%** | 98.00% |
| 7 | Cherry with healthy | **100.00%** | **100.00%** |
| 8 | Corn with grey leaf spot | **99.13%** | 96.00% |
| 9 | Corn with common rust | **100.00%** | **100.00%** |
| 10 | Corn with northern leaf blight | **99.43%** | 94.00% |
| 11 | Healthy corn | **100.00%** | **100.00%** |
| 12 | Grape with black rot | **98.51%** | 90.00% |
| 13 | Grape with black measles | **99.13%** | 98.00% |
| 14 | Grape with leaf blight | **99.54%** | 96.00% |
| 15 | Healthy grape | **100.00%** | **100.00%** |
| 16 | Orange with Huanglongbing | **100.00%** | **100.00%** |
| 17 | Peach with bacterial spot | **99.74%** | 96.00% |
| 18 | Healthy peach | **99.73%** | 98.00% |
| 19 | Pepper with bacterial spot | **99.04%** | 92.00% |
| 20 | Healthy pepper | **99.65%** | 94.00% |
| 21 | Potato with early blight | **99.45%** | 98.00% |
| 22 | Healthy potato | **100.00%** | **100.00%** |
| 23 | Potato with late blight | **99.19%** | 92.00% |
| 24 | Healthy raspberry | **100.00%** | **100.00%** |
| 25 | Healthy soybean | **99.82%** | 94.00% |
| 26 | Squash with powdery mildew | **99.32%** | 92.00% |
| 27 | Healthy strawberry | **100.00%** | **100.00%** |
| 28 | Strawberry with leaf scorch | **100.00%** | **100.00%** |
| 29 | Tomato with bacterial spot | **99.73%** | 90.00% |
| 30 | Tomato with early blight | **99.24%** | 94.00% |
| 31 | Healthy tomato | **99.92%** | 98.00% |
| 32 | Tomato with late blight | **99.76%** | 94.00% |
| 33 | Tomato with leaf mold | **99.87%** | 96.00% |
| 34 | Tomato with septoria leaf spot | **99.32%** | 96.00% |
| 35 | Tomato with two spotted spider mite | **99.45%** | 92.00% |
| 36 | Tomato with target spot | **99.54%** | 94.00% |
| 37 | Tomato with mosaic virus | **100.00%** | **100.00%** |
| 38 | Tomato with yellow leaf curl virus | **99.47%** | 96.00% |
| 39 | Background without leaf | **99.92%** | 96.00% |
| | **Average Accuracy** | **99.62%** | 96.46% |

## 5.5 Comparison with Related Works

This section compared the results obtained by the method proposed in this study with other methods found in the literature. By analyzing the comparative performance across different scenarios, it was evident that the LDPNet architecture achieved satisfactory metrics. Using Scenario 2 as the basis for training and testing, this classifier was employed as the benchmark for comparison with other related works. Table 11 presented the performance of the LDPNet architecture compared to the related works discussed in Section 3.

When compared to the work of Syamsuri and Kusuma [2019], the LDPNet model achieved substantial improvements across all reported metrics. In terms of accuracy, LDPNet achieved 99.62%, outperforming the 95.79% by 3.83 percentage points. For precision, LDPNet scored 99.16%, exceeding the 96.00% by 3.16 percentage points. Similarly, in recall, LDPNet achieved 99.08%, surpassing the 96.00% in the previous study by 3.08 percentage points. Lastly, LDPNet reached an F1-Score of 99.11%, which was 3.11 percentage points higher than the 96.00% obtained by the author.

The proposed LDPNet model demonstrated significant improvements when compared to KC *et al*. [2019], Gajjar *et al*. [2022], and Falaschetti *et al*. [2022]. In terms of accuracy, LDPNet achieved 99.62%, surpassing the 98.34% reported in KC *et al*. [2019] by 1.28 percentage points, the 96.88% in Gajjar *et al*. [2022] by 2.74 percentage points, and the 95.24% in Falaschetti *et al*. [2022] by 4.38 percentage points. Although the mentioned works did not report precision, recall, F1-Score, or AUC values.

When compared to Pandian *et al*. [2022], the LDPNet

**Table 11.** Summary of Metrics Achieved by Related Works

| Author | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | AUC (%) |
|---|---|---|---|---|---|
| Syamsuri and Kusuma [2019] | 95.79 | 96.00 | 96.00 | 96.00 | - |
| KC *et al*. [2019] | 98.34 | - | - | - | - |
| Gajjar *et al*. [2022] | 96.88 | - | - | - | - |
| Falaschetti *et al*. [2022] | 95.24 | - | - | - | - |
| Pandian *et al*. [2022] | 98.41 | 94.00 | **100.00** | 97.00 | - |
| de Sousa and de Sousa [2023] | 99.55 | 94.25 | - | 90.86 | 99.81 |
| Hashemifar and Zakeri [2024] | 98.51 | 97.00 | 98.00 | 97.00 | - |
| Present Work | **99.62** | **99.16** | 99.08 | **99.11** | **99.99** |

model showed clear advancements in most metrics. It achieved an accuracy of 99.62%, surpassing the 98.41% of Pandian *et al*. [2022] by 1.21 percentage points, and a precision of 99.16%, exceeding 94.00% by 5.16 percentage points. The F1-Score also improved from 97.00% to 99.11%, reflecting better overall performance.

Regarding recall, Pandian *et al*. [2022] achieved 100.00%, slightly higher than the 99.08% of LDPNet. This indicated superior sensitivity in detecting all positive cases in Pandian *et al*. [2022]. However, the proposed model compensated for this slight difference with higher precision, leading to fewer false positives.

The comparison between the proposed LDPNet model and the study in de Sousa and de Sousa [2023] revealed significant improvements in several key metrics. In terms of accuracy, LDPNet achieved 99.62%, surpassing the 99.55% reported in de Sousa and de Sousa [2023] by 0.07 percentage points, highlighting a slightly better overall performance. For precision, the proposed model showed a substantial improvement, reaching 99.16% compared to the 94.25% reported in de Sousa and de Sousa [2023], a remarkable gain of 4.91 percentage points, demonstrating LDPNet's superior ability to reduce false positives. The F1-Score also improved considerably, with LDPNet achieving 99.11%, significantly higher than 90.86%, reflecting a stronger balance between precision and recall. Finally, the AUC metric was slightly higher in the proposed model, achieving 99.99% compared to 99.81%.

The results presented in Table 11 showed an improvement of the proposed LDPNet model compared to the study in Hashemifar and Zakeri [2024]. In terms of accuracy, the proposed model achieved 99.62%, a gain of 1.11 percentage points over the 98.51% reported in the referenced work, indicating higher reliability in overall predictions. For precision, the LDPNet reached 99.16%, surpassing the 97.00% of Hashemifar and Zakeri [2024] by 2.16 percentage points, which demonstrated its effectiveness in minimizing false positives. The recall metric also improved from 98.00% in Hashemifar and Zakeri [2024] to 99.08%, evidencing the proposed model's stronger ability to correctly identify positive cases. Regarding the F1-Score, LDPNet achieved 99.11%, outperforming the 97.00% of the referenced study by 2.11 percentage points, which reflected the balance between precision and recall.

In summary, the comparative analysis demonstrated that the proposed LDPNet model outperformed several state-of-the-art approaches in the task of plant disease classification. Across all relevant metrics—accuracy, precision, recall, F1-Score, and AUC—LDPNet consistently achieved higher values, emphasizing its robustness and effectiveness. Notably, the model struck a remarkable balance between computational efficiency and classification performance, with significant improvements in accuracy and precision compared to methods such as those presented by Syamsuri and Kusuma [2019], KC *et al*. [2019], Gajjar *et al*. [2022], and others. While the recall metric of some prior works, such as Pandian *et al*. [2022], surpassed LDPNet, the overall performance of the proposed model remained superior due to its ability to optimize multiple metrics simultaneously. This reinforced the applicability of LDPNet in real-world scenarios, where a balanced performance across all evaluation criteria is critical. These results validated the effectiveness of the proposed approach and demonstrated its potential as a reliable solution for plant disease classification tasks.

## 5.6 Computational Efficiency and Hardware Requirements

While achieving high classification accuracy is a primary objective, the practical deployment of deep learning models in agricultural environments necessitates a rigorous evaluation of computational efficiency. In remote field conditions, diagnostic models must often execute entirely offline on resource-constrained edge devices, such as legacy smartphones or embedded systems. In these scenarios, cloud-based processing is frequently unfeasible due to intermittent or absent internet connectivity.

As previously detailed in Table 4, the proposed LDPNet architecture requires 1,817,063 parameters. This represents a reduction of 20.08% compared to MobileNetV2 (2,273,831 parameters) and 32.01% compared to the reference model from previous literature (2,672,735 parameters). Evaluating the active memory footprint required for device deployment under a 32-bit floating-point representation (4 bytes per parameter), LDPNet requires a storage size of 7.27 MB, whereas MobileNetV2 and the reference model demand 9.10 MB and 10.69 MB, respectively. This absolute reduction of 1.83 MB of active memory is a critical hardware advantage for mobile CPUs, as it significantly increases the probability that network operations can be executed within high-speed cache memory rather than relying on slower DRAM access, thereby accelerating inference time.

Furthermore, the structural compression of LDPNet translates directly into a proportional reduction in mathematical complexity. By leveraging depthwise separable convolutions within a streamlined structure, the 20.08% reduction in parameters correlates to an equivalent reduction in Multiply-Accumulate operations (MACs) and Floating Point Operations (FLOPs) per inference cycle. The evaluation of the architecture demonstrates that this level of compression yields a computational cost of 0.21 GFLOPs, enabling ultra-low inference latency and high frames-per-second (FPS) throughput on edge hardware.

Executing heavy Convolutional Neural Networks on mobile devices frequently leads to hardware overheating and subsequent thermal throttling, which can severely degrade real-time diagnostic performance. By strictly optimizing the parameter space to 1.81 million and operating within a 7.27 MB physical footprint, LDPNet mitigates excessive thermal generation and reduces power consumption. These hardware metrics confirm that LDPNet effectively satisfies the rigorous physical constraints of modern precision agriculture without compromising its outstanding 99.62% classification accuracy.

# 6 Conclusion and Future Work

This study described the proposal of a CNN architecture based on the MobileNet model, aimed at classifying leaf disease images from 13 different plant species. The proposed architecture was tested in two scenarios: with and without data augmentation, and the results were compared with MobileNetV2 and the model from the reference study.

In Scenario 1, the proposed network demonstrated superior performance across all metrics compared to MobileNetV2. The CNN achieved an accuracy of 99.26%, precision of 98.93%, recall of 98.47%, F1-Score of 98.69%, and AUC of 99.98%, highlighting its effectiveness in classifying images, even in scenarios with data imbalance.

In Scenario 2, with data augmentation, the proposed architecture outperformed MobileNetV2 in all metrics, achieving 99.62% accuracy, 99.16% precision, 99.08% recall, 99.11% F1-Score, and 99.99% AUC. Notably, the proposed CNN also outperformed the reference model from the previous study, showing significant improvements. The reference model, a deep convolutional network with a more complex architecture, achieved 96.46% accuracy, 96.47% precision, 99.89% recall, and 98.15% F1-Score.

An important point to highlight is that, despite the higher complexity and the larger number of parameters in the reference model, the proposed CNN achieved superior performance with a significantly smaller number of parameters. LDPNet has 1,817,063 parameters, while the reference model has 2,672,735 parameters. This substantial reduction in the number of parameters not only contributed to a lower computational load and greater efficiency but also demonstrated that a leaner architecture can achieve superior or equivalent performance in complex image classification tasks.

The analysis of the results confirmed that the proposed network was effective, showing a remarkable balance between performance and complexity. The introduction of data augmentation techniques further improved the model's performance, reinforcing the importance of these techniques for optimizing generalization.

Although the results were promising, there were opportunities to further enhance the approach, such as implementing and testing additional regularization techniques, like dropout and weight normalization techniques. Evaluating other advanced neural network architectures, such as attention-based networks or capsule networks, could also provide insights. Additionally, assessing the proposed CNN's performance in different application domains, such as testing the architecture in medical images for anomaly detection, could reveal how the model handled complex and variable visual features, expanding its potential application for health diagnostics. These future steps could further validate the versatility and effectiveness of the presented approach.

# Declarations

## Authors' Contributions

All authors participated in the conception of the approach and the development of the LDPNet. **Roney Nogueira de Sousa** implemented the model, conducted the experiments, and analyzed the results. **Roney Nogueira de Sousa** discussed the results with **Saulo Anderson Freitas de Oliveira** and **Pedro Pedrosa Rebouças Filho**. **Saulo Anderson Freitas de Oliveira** and **Pedro Pedrosa Rebouças Filho** supervised the project and provided guidance throughout the study. All authors participated in the writing and reviewing of the article.

**Roney Nogueira de Sousa**: Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft and editing.

**Saulo Anderson Freitas de Oliveira**: Conceptualization, Methodology, Project administration, Supervision, Validation, and editing.

**Pedro Pedrosa Rebouças Filho**: Project administration, Supervision, and Validation.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgements

The authors gratefully acknowledge the financial support provided by the Coordination for the Improvement of Higher Education Personnel (CAPES — Funding Code 001) and the Federal Institute of Education, Science and Technology of Ceará (IFCE).

## Funding

This research received funding from the Coordination for the Improvement of Higher Education Personnel (CAPES — Funding Code 001).

## Availability of data and materials

The LDPNet model developed and analyzed during the current study is not publicly available due to restrictions related to ongoing research. However, it may be made available upon reasonable request to the corresponding author.

# References

Angelotti, F., Hamada, E., and Bettiol, W. (2024). A comprehensive review of climate change and plant diseases in brazil. *Plants*, 13(17). DOI: 10.3390/plants13172447.

Bharate, A. A. and Shirdhonkar, M. S. (2017). A review on plant disease detection using image processing. In *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, pages 103–109. DOI: 10.1109/ISS1.2017.8389326.

de Oliveira, C. M., dos Santos Pereira, T., Mouro, L. P., and Canale, M. C. (2026). A disease complex: crop losses and economic impact of corn stunt diseases on brazilian corn production. *Crop Protection*, 202:107549. DOI: 10.1016/j.cropro.2026.107549.

de Sousa, R. N. and de Sousa, R. N. (2023). Development of a convolutional neural network architecture for classifying foliar diseases in plants. In *Anais do XVI Encontro Unificado de Computação do Piauí*, pages 1–8, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/enucompi.2023.26610.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7). Available at:`https://jmlr.org/papers/v12/duchi11a.html`.

Falaschetti, L., Manoni, L., Di Leo, D., Pau, D., Tomaselli, V., and Turchetti, C. (2022). A cnn-based image detector for plant leaf diseases classification. *HardwareX*, 12:e00363. DOI: 10.1016/j.ohx.2022.e00363.

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and electronics in agriculture*, 145:311–318. DOI: 10.1016/j.compag.2018.01.009.

Food and Agriculture Organization of the United Nations (FAO) (2024). The hidden health crisis: How plant diseases threaten global food security. *FAO One Health Highlights*. Available at:`https://www.fao.org/one-health/highlights/how-plant-diseases-threaten-global-food-security`.

Funck, F. C. (2019). Detectando a ferrugem asiática na folha da soja utilizando redes neurais convolucionais. B.S. thesis, Universidade Tecnológica Federal do Paraná. Available at:`http://repositorio.utfpr.edu.br/jspui/handle/1/15628`.

Gai, Y. and Wang, H. (2024). Plant disease: A growing threat to global food security. *Agronomy*, 14(8). DOI: 10.3390/agronomy14081615.

Gajjar, R., Gajjar, N., Thakor, V. J., Patel, N. P., and Ruparelia, S. (2022). Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform. *The Visual Computer*, pages 1–16. DOI: 10.1007/s00371-021-02164-9.

Geetharamani, G. and Pandian, A. (2019). Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Computers & Electrical Engineering*, 76:323–338. DOI: 10.1016/j.compeleceng.2019.04.011.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press. DOI: 10.1016/c2024-0-03581-3.

Hashemifar, S. and Zakeri, M. N. (2024). Deep identification of plant diseases. In *2024 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pages 1–6. DOI: 10.1109/AISP61396.2024.10475267.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861. DOI: 10.48550/arxiv.1704.04861.

Huber, D. M. and Jones, J. B. (2013). The role of magnesium in plant disease. *Plant and soil*, 368:73–85. DOI: 10.1007/s11104-012-1476-0.

Husson, O., Sarthou, J.-P., Bousset, L., Ratnadass, A., Schmidt, H.-P., Kempf, J., Husson, B., Tingry, S., Aubertot, J.-N., Deguine, J.-P., *et al*. (2021). Soil and plant health in relation to dynamic sustainment of eh and ph homeostasis: A review. *Plant and Soil*, 466(1):391–447. DOI: 10.1007/s11104-021-05047-z.

KC, K., Yin, Z., Wu, M., and Wu, Z. (2019). Depthwise separable convolution architectures for plant disease classification. *Computers and Electronics in Agriculture*, 165:104948. DOI: 10.1016/j.compag.2019.104948.

Khirade, S. D. and Patil, A. B. (2015). Plant disease detection using image processing. In *2015 International conference on computing communication control and automation*, pages 768–771. IEEE. DOI: 10.1109/iccubea.2015.153.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. DOI: 10.48550/arxiv.1412.6980.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90. DOI: 10.1145/3065386.

Li, Z., Yang, Y., Li, Y., Guo, R., Yang, J., and Yue, J. (2020). A solanaceae disease recognition model based on se-inception. *Computers and Electronics in Agriculture*, 178:105792. DOI: 10.1016/j.compag.2020.105792.

Lu, Y., Yi, S., Zeng, N., Liu, Y., and Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, 267:378–384. DOI: 10.1016/j.neucom.2017.06.023.

MOININA, A., LAHLALI, R., and BOULIF, M. (2019). Important pests, diseases and weather conditions affecting apple production in morocco: Current state and perspectives. *Revue Marocaine des Sciences Agronomiques et Vétérinaires*, 7(1). Available at:`https://www.researchgate.net/publication/325973102_Important_pests_diseases_and_weather_conditions_affecting_apple_production_current_state_and_perspectives`.

Pandian, J. A., Kanchanadevi, K., Kumar, V. D., Jasińska, E., Goňo, R., Leonowicz, Z., and Jasiński, M. (2022). A five convolutional layer deep convolutional neural network for plant leaf disease detection. *Electronics*, 11(8). DOI: 10.3390/electronics11081266.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

DOI: 10.1109/cvpr.2018.00474.

Sousa, R., Oliveira, S., and Filho, P. R. (2024). Desenvolvimento de um modelo convolucional leve para identificação de doenças foliares. In *Anais Estendidos da XXXVII Conference on Graphics, Patterns and Images*, pages 117–122, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/sibgrapi.est.2024.31655.

Syamsuri, B. and Kusuma, G. P. (2019). Plant disease classification using lite pretrained deep convolutional neural network on android mobile device. *Int. J. Innov. Technol. Explor. Eng*, 9(2):2796–2804. DOI: 10.35940/ijitee.b6647.129219.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9. DOI: 10.1109/cvpr.2015.7298594.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning. In *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 270–279, Cham. Springer International Publishing. DOI: 10.1007/978-3-030-01424-7$_2$7.

WHO, W. H. O. (2024). Risco de fome generalizada aumenta pelo quinto ano consecutivo. Available at:`https://news.un.org/pt/story/2024/04/1830741`.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76. DOI: 10.1109/jproc.2020.3004555.