Journal of Plant Diseases and Protection (2024) 131:1043–1059
https://doi.org/10.1007/s41348-024-00915-z

ORIGINAL ARTICLE

LWDN: lightweight DenseNet model for plant disease diagnosis

Akshay Dheeraj[1,2] · **Satish Chand**[2]

Received: 5 October 2023 / Accepted: 21 March 2024 / Published online: 12 April 2024

Abstract

Plant disease diagnosis in smart agriculture is a crucial issue that carries substantial economic significance on a global scale. To address this challenge, intelligent and smart agricultural solutions are currently being developed to assist farmers in implementing preventive measures to increase crop production. As deep learning technology continues to evolve, many convolutional neural network (CNN) models have emerged as highly effective for detecting plant leaf diseases. These CNN-based models require heavy computation and processing cost. So, this paper develops a new lightweight deep convolutional neural network named lightweight DenseNet (LWDN) for detection of plant leaf disease for agricultural applications. Based on the DenseNet121 architecture, the presented model comprises pruned and concatenated architecture of DenseNet121. The presented study involved training and testing a proposed model (LWDN) on the PlantVillage dataset to acquire a knowledge of plant disease features. The model was trained using a combination of partial layer freezing, transfer learning, and feature fusion techniques. Out of several models experimented with, the proposed model has 99.37% classification accuracy, a model size of 13.8 MB, with 1.5 M parameters. The proposed model has 93% fewer parameters than InceptionV3 and Xception and 90% and 50% fewer parameters compared to VGG16 and MobileNetV2, respectively. Furthermore, the proposed method has superior diagnostic capabilities compared to several prior studies and larger state-of-the-art models utilizing plant leaf images. The compact size and competitive accuracy of the LWDN model render it appropriate for real-time plant diagnosis on portable and mobile devices with restricted computational resources.

Keywords PlantVillage · Pruning · Concatenation · Lightweight · Diagnosis

Introduction

Plant disease is a major threat that affects the global food supply and threatens food security. This affects agricultural production and quality and increases the economic loss to farmers. Pest attacks result in crop losses ranging from 10 to 40% on a global scale every year (Savary et al. 2019). Hence, timely recognition of plant diseases is crucial. The plant leaves are usually examined to detect these diseases. Nevertheless, many farms and plantations still rely on traditional methods to detect plant diseases with the naked eye, which are time-consuming, laborious and need incessant monitoring and result in high expenses and inaccuracies in large farms. Automated approaches for identifying plant disease are preferred over manual methods because of the limitations of human perception in detecting plant leaf diseases of all types. Manual identification techniques are also prone to errors, time-consuming, and only feasible for small areas (Tiwari et al. 2021).

The advancement in digital cameras and artificial intelligence techniques has brought a significant transformation to the field of plant leaf disease detection and enhanced cultivation productivity (Jackulin and Murugavalli 2022; Thakur et al. 2022). Machine learning models have been extensively employed in the purpose of plant disease diagnosis for the last two decades. Dubey and Jalal (2016) presented a support vector machine-based approach for apple disease and achieved an accuracy of 95.94%. The conventional vision-based methods generally use manual feature extraction, which is a laborious and expensive affair. Most machine

✉ Akshay Dheeraj
akshaydheeraj.jmi@gmail.com; akshay.dheeraj@icar.gov.in

Satish Chand
schand20@gmail.com

1 ICAR-Indian Agricultural Statistics Research Institute, New Delhi, India

2 School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India

learning algorithms perform poorly and yield unsatisfactory results when the dataset is large.

In the past decade, convolutional neural networks (CNNs), a specific category of deep learning algorithms, have effectively addressed the challenges of object detection and classification, overcoming the restrictions of conventional machine learning techniques (Ferentinos 2018). Recently, CNNs have become the most extensively used model for detecting plant leaf disease with automatic feature extraction with minimal effort (Joshi et al. 2021; Yu et al. 2023). Plant leaf disease diagnosis has been accomplished by the EfficientNet model and by utilizing PlantVillage dataset (Atila et al. 2021). The proposed study, despite having a smaller parameter count, exhibited superior performance in terms of average accuracy when compared to VGGNet16, RestNet50, Inception-V3 and AlexNet models. Hanh et al. (2022) utilized EfficientNet B3 and EfficientNet B5 architectures to enhance the disease identification performances for plants. The proposed EfficientNet B3 and EfficientNet B5 models obtained 99.997% accuracy on original and augmented PlantVillage dataset. In another study, Tiwari et al. (2021) developed a disease recognition framework for six crops with 27 diseases, using DenseNet201 architecture and attained an average performance accuracy of 99.19%. In another work, a novel approach for capturing subtle features of lesions in plant images was proposed by integrating MobileNetV2 with soft attention (Chen et al. 2021a). This technique achieved remarkable accuracy results of 99.13% and 99.71% for the local and PlantVillage dataset, respectively.

Several lightweight models have also been presented recently for plant disease diagnosis (Xiao et al. 2023; Liu et al. 2023; Chen et al. 2021b). Sharma et al. (2023) developed a lightweight deep CNN model for plant disease recognition comprising a sequence of collective blocks. The model proposed here attained a classification score of 95.49% with 6.4 million parameters. Researchers have recently focused on encoder–decoder network architectures and image fusion techniques for the precise and prompt identification of several plant diseases (Udendhran and Balamurugan 2021). Fan et al. (2022) presented a technique using feature fusion and transfer learning for plant disease recognition. A feature fusion approach was employed to combine deep and handcrafted features to extract more relevant information from leaf images. The developed model yielded an average accuracy score of 96.5%. Many CNN models with feature fusion have been proposed for medical imaging analysis, offering remarkable accuracy with comparatively fewer number of parameter and lesser computational complexity and cost (Montalbo 2021, 2022). The experiment results show that these CNN models with feature fusion perform satisfactorily.

However, different other approaches may still result in enhancements that decrease both computational expenses and optimization requirements while achieving better accuracy. Despite advancements in the field, the research community continues to encounter the challenge of developing an efficient and lightweight model with fewer parameters and an appropriate model size for practical agricultural implementations. The rationale behind the need for a lightweight model in agricultural applications is due to the fact that complex model architectures having a significant number of parameters may experience high-bias issues, which can hinder their capability to fit the training data effectively.

This research study is aimed to enhance classification accuracy while utilizing a relatively compact model. Therefore, this research study presents a lightweight CNN model utilizing different techniques like model pruning, freezing some layers, and feature fusion with DenseNet121 on the PlantVillage dataset (augmented one) to recognize and categorize diseases of plants. The key primary contributions of the proposed research study have been summarized as follows:

- This study pruned a pre-trained DenseNet121 model. In the pruned DenseNet121 model, parameters count and network size are reduced, reducing the complexity of the network without drastically affecting the performance. This pruned DenseNet121 is trained faster because of reduced network size along with maintaining the rich extraction of relevant features.
- We constructed an integrated model by fusing replicated pruned DenseNet121 model where the replicated one has been fully retrained from PlantVillage and ImageNet dataset to generate the feature set. Some upper layers have been frozen to generate different feature sets in another half of the model. The feature set of these two models (original and replicated one) was fused and sent to another set of additional layers. Different feature set have been generated through the concatenation of these pruned models.
- Precision, recall, F1-score, and accuracy have been employed as performance metrics for assessing the performance of the proposed work. Additionally, the results of the experimental study have been compared against seven classical CNN models, namely DenseNet121, XceptionNet, InceptionV3, VGGNet-16, MobileNetV2, EfficientNet B0, and NasNetMobile.
- Compared to the existing Plant disease identification work on the PlantVillage dataset, the proposed work achieved competitive results with 1.5M parameters.
- Statistical analysis of experimental results has been done using the Friedman test. Upon evaluation, results

show that all the demonstrated models significantly differ when executed on the PlantVillage dataset.

The experimental findings indicate that the developed model has remarkable classification accuracy, while being computationally less complex and cost-effective. Therefore, the proposed methodology has the potential to be readily deployable, upgradable and, most importantly, applicable to future applications.

The remainder of this paper is organized as mentioned: The related work section presents existing literature work outcomes published in the domain. Materials and methods section elaborates the proposed model. Experiment study and results section present the performance assessment of the proposed study. At last, the research article is concluded and presented in Conclusion section.

Related work

In the last decade, plant disease diagnosis using image processing techniques and deep learning has been a prominent area of research. Conventional machine learning involves the feature extraction process and encompasses various features such as colour, shape, texture, and vein. This is achieved by utilising techniques such as histogram, Haar, SURF, LBP, GLCM, SIFT, Fourier transform, Gabor filter, curvelet, wavelet, and graph representations (Sachar and Kumar 2021). Plant disease identification has been performed using machine learning algorithms, including KNN, SVM, and the Ensemble Tree. Previous studies have considered multiple features, including shape, colour, and texture features. Shrivastava and Pradhan (2021) used colour features to classify rice disease with an image dataset of 619 images. The authors extracted 172 different colour features and used SVM and attained 94.6% accuracy. Table 1 demonstrates that the utilization of manual machine learning techniques results in accuracy ranging from 85.7% to 99.10%. The research study by Prajapati et al. (2017) utilized shape, colour, and texture features to classify three diseases of rice crops and used k-means algorithm for segmentation of relevant infected disease portions. The authors used their own rice disease dataset of 120 images, extracted 88 distinct features, and yielded 88.57% accuracy using SVM classifier. Chuanlei et al. (2017) presented the apple disease classification system. The authors extracted 37 different features from the segmented images. These features were selected using genetic algorithm and CFS, and then SVM was employed as a classifier to detect three leaf diseases of apples. The presented model is less complex because of the utilization of a lesser number of features, with 94% accuracy rate. Zhang et al. (2017) employed a sparse representation classification algorithm after extracting shape and colour characteristics from lesions to classify cucumber leaf diseases of seven different types. In another study, local binary patterns were utilized to extract features and employed a one-class classifier to distinguish between diseased and healthy leaves in crops (Pantazi et al. 2019). Recent research used multiple features like colour histograms, Hu Moments, Haralick, and LBP for feature extraction, followed by different algorithms to classify tomato diseases (Basavaiah and Anthony, 2020). In another study, for feature extraction, fractional-order Zernike moments (FZM) was used, and SVM was employed for disease identification in grape leaf (Kaur et al.

Table 1 Traditional machine learning-based studies for plant disease identification

Authors	Datasets	Technique	Features taken	Accuracy
Shrivastava and Pradhan (2021)	Rice dataset with 619 images	SVM	Colour feature (172 features)	94.6%
Chuanlei et al. (2017)	Apple dataset	SVM	Colour, shape, and texture	94%
Zhang et al. (2017)	Own	Sparse representation classification algorithm	Shape and colour features	85.7%
Prajapati et al. (2017)	Own	SVM	Shape, colour, and texture (88 features)	88.57%
Pantazi et al. 2019	Own	SVM	Texture and colour	95%
Kaur et al. (2019)	Grape	SVM	FZM	97.34%
Basavaiah and Anthony (2020)	Tomato disease with 500 images	Random forest and Decision tree	Hu moments, colour histograms, Haralick, and LBP	Decision Tree = 90% Random Forest = 94%
Kumar et al. (2018)	PlantVillage	Exponential SMO + SVM	82 features extracted by SPAM	92.12%
Mustafa et al. (2020)	LeafSnap, Flavia, and ICL	Hybrid SVM + PNN + NB + FIS	Binarization and individual level discrete 2D wavelet	99.10%
Kurmi et al. (2021)	Pepper, potato, and tomato	SVM	SIFT	94.35%

2019). The author used a dataset comprising 400 images and yielded a 97.34% accuracy. Kurmi et al. (2021) used SVM to classify the disease associated with common pepper, potato, and tomato. Discriminative features were generated using Fisher vectors, which involved multiple-order differentiation of Gaussian distribution. The study reported 94.7% accuracy. Mustafa et al. (2020) employed a hybrid technique for early disease associated with ten different herb species. The author utilized odour extraction using an electronic nose for analysing odour and used it with shape, colour, and texture features, yielding a classification score of 99.10%. Kumar et al. (2018) utilized subtractive pixel adjacency matrix and extracted 686 features, further reducing these features (82 features) by the spider monkey optimization algorithm. These reduced features were then sent to SVM, which reported 92.12% accuracy.

However, conventional machine learning, which involves extracting features manually, is afflicted with limitations stemming from its computational complexity and significant energy consumption (Sachar and Kumar 2021). A recent research study for classifying diseases in plants using machine learning is summarized in Table 1.

Using deep learning approaches with automated feature extraction in crop disease diagnosis research continues to surpass the performance of traditional machine learning approaches (Turkoglu et al. 2022). Convolutional neural networks (CNNs) are the most commonly proposed deep learning methods for diagnosing disease using plant leaf images. Numerous other deep learning models have also been suggested for this task. Table 2 demonstrates the effective application of several CNN models for plant disease diagnosis, utilizing various plant datasets. These models have achieved remarkable accuracies exceeding 99%. Plant disease diagnosis research has been done on the PlantVillage dataset majorly (Mohanty et al. 2016; Shoaib et al. 2023). Dheeraj and Chand (2023) utilized EfficientNet B0 model to detect diseases of pepper, potato and tomato. Their work achieved 99.79% accuracy score. Gokulnath (2021) proposed a fusion approach with CNN for identifying diseases in plants. The proposed LF-CNN model achieved 98.83% accuracy. Error rate in loss function was reduced by the fusion approach, which enhanced the accuracy score of the model. Nigam et al. (2023) developed a disease classification system for three rust diseases associated with wheat. Their method used EfficientNet B4 model and attained 99.35% accuracy on their own dataset of 6556 images. Ensemble-based CNN models have also been developed. Turkoglu et al. (2022) utilized an ensemble of six CNN models and achieved 97.56% accuracy score. In this research study, authors created their own dataset named Turk-Plant dataset, having 15 types of disease and a total of 4447 images. In another study, an ensemble model was developed by employing MobileNetV2 and Xception. The study concatenated the features extracted by these two models and yielded 99.10% performance accuracy (Sutaji and Yıldız, 2022). However, it should be noted that ensemble models are more time-consuming and result in larger model file sizes because of their massive parameters. Karthik et al. (2023) proposed a coffee disease identification system where authors used Inception module with multihead attention mechanism to extract a complex pattern from the leaf images. Filters of various sizes were used at many scales and abstraction levels.

Research work by Kaya and Gürsoy (2023) employed the image fusion approach where both RGB and segmented images were used as dual input, and DenseNet121 was utilized as a classification model. The experimental study obtained 98.17% average accuracy when executed on PlantVillage. Various CNN models with feature fusion approaches were also presented for plant disease recognition (Yang et al. 2021; Fang et al. 2022; Zhang et al. 2022). Fan et al. (2022) introduced a feature fusion and transfer learning-based approach for plant leaf disease classification, which yielded an average performance accuracy of 99.5% across three different datasets. The majority of the work using CNN with a feature fusion approach was done on own data rather than on the standard PlantVillage dataset. Table 2 provides literature work on CNN models used for the classification of plant diseases along with the dataset used and accuracy results.

Comparing techniques presented in the literature is a difficult task because of variations in the datasets used. Furthermore, some techniques focus on classifying solely four plant diseases, while others are designed to classify over 38 types of plant diseases. While some models presented in the literature exhibit high performance, their complex network architecture and computationally expensive nature represent a significant drawback.

Referring to previous research work, the current study concentrates specifically on the DenseNet121 architecture, which has been modified and named Lightweight DenseNet121 (LWDN). The LWDN has been utilized for extraction of features and categorization of diseases in the current study. The advantage of the LWDN is that it provides competitive performance for plant disease identification while employing a reduced number of parameters than classical CNN models. Additionally, this research study entails a comparative evaluation of the proposed model with seven classical CNN models to determine its performance.

Table 2 Related work using CNNs for plant disease identification

Research work	Dataset	Model/technique	Accuracy
Mohanty et al. (2016)	PlantVillage	AlexNet and GoogleNet	99.35%
Ramcharan et al. (2017)	Cassava leaf disease dataset	Inception v3	96.6
Liu et al. (2017)	Apple leaf diseases	AlexNet	97.62%
Ma et al. (2018)	Cucumber PlantVillage	Custom CNN	93.4%
Too et al. (2019)	PlantVillage	DenseNet	99.75%
Kamal et al. (2019)	PlantVillage	MobileNet	98.34%
Chen et al. (2020a)	Apple, grape, potato (PlantVillage), own maize, and rice disease dataset	MobileNetV2	99.94% on PlantVillage 99.85% on own dataset
Chen et al. (2020b)	PlantVillage maize, own rice and maize	VGGNet with Inception modules	92% on PlantVillage Maize, 80.38% custom maize and 92.00 on own rice dataset
Jiang et al. (2020)	Own rice leaf disease dataset	Customized CNN (feature extractor) + SVM (classifier)	96.8%
Abbas et al. (2021)	PlantVillage Tomato	DenseNet121 with C-GAN	94.34% and 97.11% on PlantVillage and PlantVillage with synthetic images dataset, respectively
Atila et al. (2021)	PlantVillage	EfficientNet (B0–B7)	99.91% and 99.97% on EfficientNet B5 and B4, respectively for original and augmented dataset
Shin et al. (2021)	Custom strawberry dataset	ResNet-50	98.11%
Yang et al. (2021)	Custom citrus	ResNet with feature fusion and attention mechanism	97.89%
Gokulnath (2021)	PlantVillage	LF-CNN	98.93%
Turkoglu et al. (2022)	Turk-Plants	For feature extraction, the Ensemble model used and SVM was used as classifier	97.56% and 96.83% for MV and EF model, respectively
Fan et al. (2022)	Coffee and two apple leaf	InceptionV3 (deep features) with handcrafted features	97.12% on coffee, 92.59% and 99.79% on two apple datasets
Naik et al. (2022)	PlantVillage and own chilli leaf dataset	SECNN	99.28% on combined dataset
Sutaji and Yıldız (2022)	PlantVillage and own dataset	Ensemble MobileNetV2 and Xception	99.10% on PlantVillage, 99.52% on own dataset
Kaya and Gürsoy (2023)	PlantVillage	Multi-headed DenseNet	98.17%
Nigam et al. (2023)	Own wheat dataset	EfficientNet B4	99.35%
Dheeraj and Chand (2023)	PlantVillage	EfficientNet B0	99.79%
Karthik et al. (2023)	BRACOL	Inception module with attention mechanism	98.57%

Materials and methods

This section presents a description of the proposed model and a comprehensive overview of the dataset used in this study.

Dataset

The proposed research study utilizes the PlantVillage augmented dataset (https://data.mendeley.com/datasets/tywbtsjrjv) to classify images based on plant species and their associated diseases. The PlantVillage dataset comprises 39 categories, including background images, encompassing fourteen plant species exhibiting distinct plant diseases. A total of 14 distinct plant species were examined, among which 17 categories were found to exhibit fungal diseases, while four were identified as having bacterial diseases. Additionally, two species were observed to have viral diseases, two were found to be affected by fungal diseases, and the remaining one was affected by mite-induced disease. The healthy category of plant leaves contains 12 different plant species, and the total number of images is 61,486. Each image comprised R, G, and B channels and had a size of 256 × 256. The whole collection of 61,486 images had properly distributed to train, validation, and test in the ratios 80%, 10%, and 10%, respectively.

Method overview

This section covers the developmental process of the proposed model for identifying plant diseases. It includes a background of DenseNet121 architecture which serves as the base architecture, and some added structure to the DenseNet121 architecture to minimize the number of parameters and mitigate overfitting while attaining a reasonable performance level.

DenseNet121

Convolutional neural networks (CNNs) have garnered widespread attention as a promising solution for image classification tasks, specifically in the detection of plant diseases. Several pre-trained models are available to facilitate the classification of various image types. Furthermore, transfer learning has also been utilized for image classification applications by leveraging pre-trained models. The present approach involves modifying the top layers of pre-trained models to enable the classification of novel image categories. Given the diversity of plant leaf characteristics, a pre-trained neural network can effectively address this challenge, making transfer learning (TL) an area of focus for researchers in plant disease detection (Kılıç and Inner 2022). Several pre-trained convolutional neural network (CNN) models have been proposed for image classification tasks (He et al. 2016; Brahimi et al. 2017; Alom et al. 2018; Alzubaidi et al. 2021; Uğuz and Uysal 2021; Pandey and Jain 2022). However, amidst these models, DenseNet outperforms the rest in recognition accuracy and computation time by utilizing substantially fewer amount of parameters. (Singh et al. 2019).

DenseNet is one of the deep learning architectures that enable efficient propagation of information by providing direct access to loss function and gradient to each layer, thereby enhancing the depth of training (Huang et al. 2017). This is achieved by interconnecting all layers in a feed-forward manner, where all layers are densely connected to each other, unlike ResNet (He et al. 2016). DenseNet merges image features by utilizing a concatenation operator, and it employs considerably fewer parameters compared to other CNN models. DenseNet comprises various architectures, namely DenseNet-121, DenseNet-160, and DenseNet-201, with each architecture having a distinct number of layers. In this study, DenseNet-121 was chosen, which comprises [5 + (6 + 12 + 24 + 16) × 2) = 121] layers and possesses a moderate count of trainable parameters.

- Transition layers: Three (6, 12, 24)
- Pooling and Folding layers: Five
- Classification layer: Sixteen
- Dense blocks: 2(1 × 1 and 3 × 3 conv)

Convolutional neural networks generate the lth output layer by performing a nonlinear transformation, H_l, on the output of the preceding layer, X_{l-1} (Huang et al. 2017). In contrast, DenseNet concatenates the output features of the layers with the input features instead of adding them together (Zhang et al. 2019). The DenseNet architecture enhances the propagation of information between layers by granting direct access to the feature maps of all previous layers as inputs to the l^{th} layer. The mathematical expression of the operation is as follows:

$$X_l = H_l\left(X_0, X_1, X_2, X_3, .., X_{l-1}\right) \tag{1}$$

The concatenation of the output maps of previous layers is represented as a tensor $X_0, X_1, X_2, X_{3,\ldots\ldots}, X_{l-1}$ in Eq. (1), while H_l represents a nonlinear transformation function (Cai et al. 2021). The function H_l comprises four primary components: pooling, activation function (ReLU), convolution and batch normalization. The increase rate k is used for enhancement of the generalization ability of the lth layer. The value of k is defined as:

$$k[l] = (k[0] + k(l—1)) \tag{2}$$

Here, *k [0]* represents the initial channel number.

The DenseNet model has input, dense, and transition blocks. Input blocks have a convolutional layer of 7 × 7 followed by batch normalization (BN), rectified linear unit (ReLU) and Max pooling layer of dimension 3 × 3. After the input block, dense blocks are there with BN, ReLU, and 1 × 1 convolutional, followed by another set of BN, ReLU and 3 × 3 convolutional layer. The transition block is composed of BN, ReLU, 1 × 1 convolutional, followed by an average pooling layer of dimension 2 × 2. The DenseNet architecture differs from other deep learning models, such as residual networks, that rely on feature summation and have a large number of parameters. Instead, DenseNet incorporates dense blocks with a growth rate of k that are concatenated to every layer of the network. This technique enables efficient end-to-end propagation of feature inputs from preceding layers to succeeding layers, facilitating the propagation of high-quality gradients even at bigger depths while maintaining a relatively less amount of parameters.

This makes it a suitable choice for the task at hand. Similar to other deep convolutional neural network (DCNN) models, the DenseNet architecture includes a downsampling layer to avoid resource depletion during feature extraction. Specifically, it incorporates a transition layer that uses 1 × 1 convolution and a 2 × 2 average pooling operation with strides of 1 and 2 for dimensionality reduction in feature maps. This aids in maintaining computational efficiency during training and inference.

The proposed LWDN architecture

Traditional convolutional neural networks (CNNs) typically adopt a strategy for stacking multiple convolutional layers to improve performance results. The complexity of the computation and the parameters count both rise with this method. Despite possessing considerably fewer parameters than most DCNNs, the DenseNet model remains subject to high computational demands. In this regard, the proposed method seeks to reduce both the parameters and computation complexity of the DenseNet121 model while preserving its performance. Considering the main objective of DenseNet, which is to process large datasets like ImageNet containing over 1000 categories and 14 million images, replicating and training this proposed model is challenging due to the constraints of the available computational resources. Furthermore, employing the entire model's structure for the limited dataset at hand only contributes to increased complexity and resource consumption. Therefore, through a proposed model pruning, model concatenation and feature fusion technique, a model named lightweight DenseNet121 (LWDN) has been created.

Model pruning and concatenation technique

To reduce the parameter size and computational complexity of DenseNet121, we applied a pruning technique to remove a significant number of layers from the architecture. This resulted in a shortened end-to-end structure that is more efficient in terms of computational resources while still getting an adequate level of performance accuracy in plant disease classification. Generally, pruning performs layer reduction which eventually reduces the parameter count and size of the architecture (Das et al. 2020). Figure 1 shows the proposed model pruning technique where six dense blocks are there, followed by a transition layer. This transition layer is then connected to another set of four dense blocks. The pruned model, referred to as the Lightweight DenseNet1 model (LWDN_1), significantly reduced the amount of parameters and depth of the original DenseNet121 model. The pruned model retains the original architecture but with a reduced number of parameters, enabling efficient training and deployment. Initially, the DenseNet121 architecture has 8 M parameters and a network length of 430, whereas the pruned LWDN_1 architecture has a parameter amount of

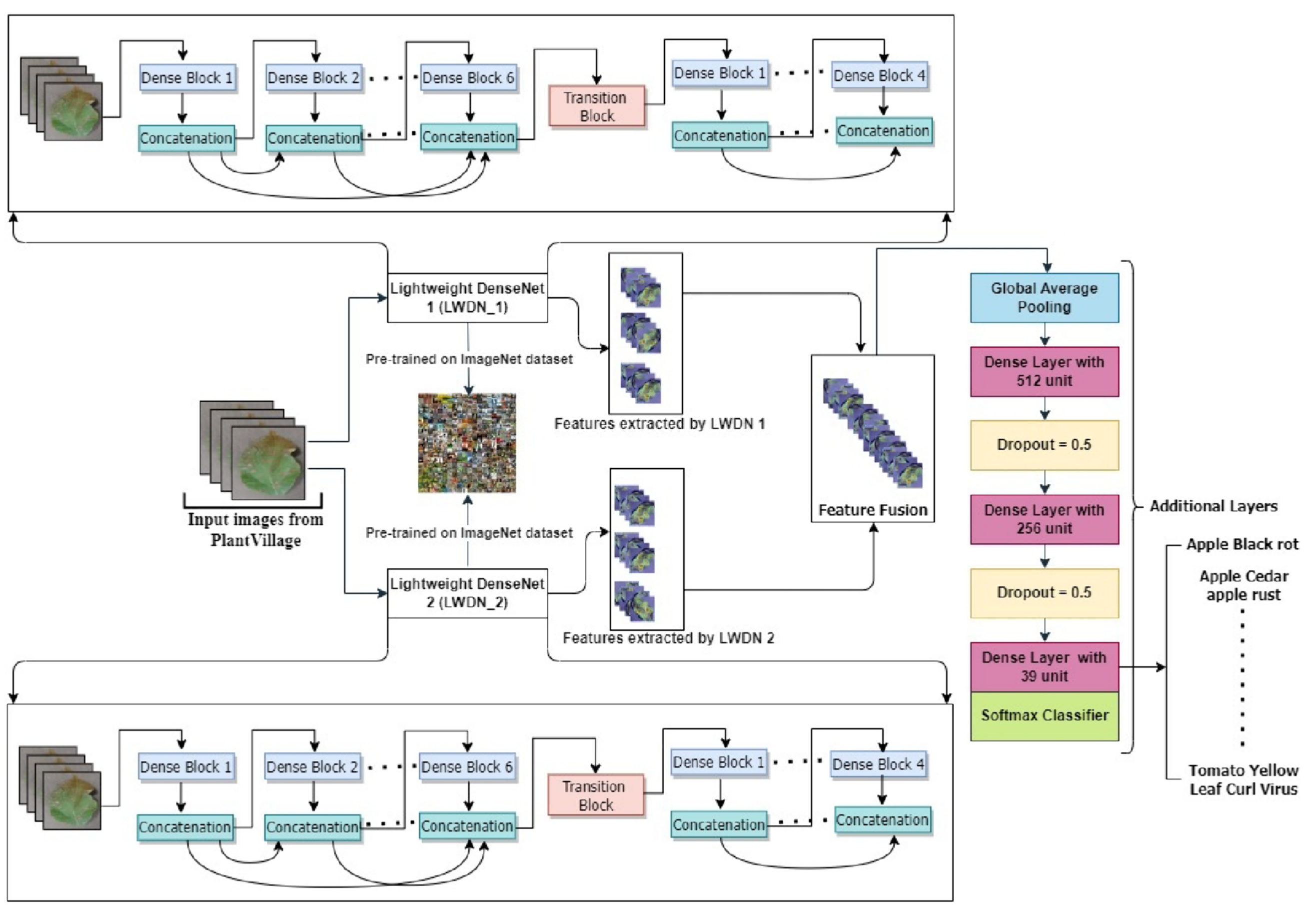

Fig. 1 Architecture of the proposed LWDN model

624 k and a network length of 81. Thus, the parameter size has been decreased by 92 per cent.

The LWDN_1 model has a lesser number of parameters due to its reduced network complexity. However, when trained on the PlantVillage dataset, it has comparatively lower performance than other classical CNN models, as shown in the ablation study. Further pruning of the network does not enhance the performance, as described in the ablation study. It is noteworthy that pruning results in a network with fewer layers, facilitating quicker weights propagation during training, and saving a substantial amount of computing resources. However, in the present study, this advantage is accompanied by a significant disadvantage as well. Reducing the number of layers for feature generation or extraction ultimately leads to decreased model performance due to the limited trainable parameters when compared to the base DenseNet121 model. To overcome this challenge, the proposed method incorporated a feature fusion and model concatenation approach (Montalbo 2021).

To extract the relevant feature to enhance the performance, we replicated the LWDN_1 architecture, named LWDN_2, and then these two were combined together to form lightweight DenseNet (LWDN). Through the feature fusion approach, features are fused together to give a robust feature set. The proposed study added a set of additional layers consisting of global average pooling or GAP (Kamal et al. 2019), a dense layer with 512 units utilizing ReLU as an activation function, dropout with a value of 0.5, succeeded by a dense layer with 256 units and ReLU activation, and dropout with value 0.5 connected with another dense layer (Dahl et al. 2013) with thirty-nine units with Softmax classifier (Fu et al. 2022). The advantage of global average pooling (GAP) in dense neural networks is that it provides a more compact and interpretable feature representation for each class. Instead of flattening the feature maps into a vector and passing it through a fully connected layer, GAP averages each feature map channel-wise. It returns a single value for each channel. Furthermore, the ReLU activation function employed in the layer introduces nonlinearity to the network and restricts output values to binary (1 or 0), thus increasing efficiency and reducing the computational cost. Additionally, the dense layer with Softmax function comprises only thirty-nine neurons, each corresponding to one class of interest. The output values from this layer represent the probability that a given input image belongs to each of the different classes. The Softmax function applied in this layer normalizes these probabilities, ensuring that they sum up to 1.0, thereby making it easier to interpret the results as probabilities. The purpose of incorporating additional layers is to enhance model performance and mitigate overfitting concerns (Bevers et al. 2022). The architecture of the proposed LWDN model is shown in Fig. 1.

Different technique for training pruned network

The utilization of certain techniques can mitigate the issue of reduced trainable parameters and performance resulting from pruning. However, deploying similar models may result in feature redundancy and a corresponding increase in computational costs without significant improvement. As a remedy, this study proposes the use of diverse methods to train each pruned model and generate a range of distinctive features. Specifically, the proposed scheme employs fine-tuning and partial layer freezing techniques to address the aforementioned issues. By using these techniques, the model can generate a range of distinctive features that can aid in improving overall performance.

Initially, both LWDN_1 and LWDN_2 models leverage their image recognition capability by transfer learning approach and learn the features from the ImageNet dataset, thus improving their performance on the plant disease detection task. Following this pre-training phase, the models undergo fine-tuning and partial layer freezing to facilitate the classification of plant diseases. In this study, partial layer freezing refers to the setting of a model's layers to a frozen state so that the pre-trained weights obtained from ImageNet are preserved from being overwritten during training (Isikdogan et al. 2020). Only the concatenation layer and the proposed set of ending layers are updated during training. The concept of layer freezing is a technique derived from fine-tuning, which enables adjustment of pre-trained weights by model towards the newly added ending layers, thereby effectively solving particular tasks (Montalbo 2021). However, when the same technique is applied to the other model, outputs are generated with no contribution to the feature set. As a contrasting approach, the other model's layers were set to an unfrozen state, allowing new weights to flow and generate diverse features throughout its entire network. Specifically, in one of the LWDN models, LWDN_1, its layers were frozen, while in LWDN_2, all layers were retrained using ImageNet and PlantVillage datasets to generate a distinct feature set. As a result, the proposed approach, which combines fine-tuning and new weights re-initialization, resulted in a broad range of diverse features being curated.

Fine-tuning hyper-parameters

Before the commencement of the training process, hyper-parameters and a loss function are chosen for the model. The hyper-parameters denote the configurable components of a deep learning model, which can significantly influence its learning process and cannot be modified during training (Yu and Zhu 2020). A loss function was also incorporated to calculate and minimize errors during both the training and validation stages. Optimal selection of hyper-parameters and loss function plays a critical role in achieving efficient results. Notably, unlike other studies, no rigorous optimization techniques were employed in this work for hyper-parameter fine-tuning. This approach demonstrated the model's adaptability and reproducibility with the dataset.

Learning rate (LR), optimizer, batch size (BS), loss function, epochs, and dropout rate (DR) are the tuned hyper-parameters of the model. These hyper-parameters of the model have been defined in Table 3. The learning rate is set to be 0.0001, and batch size has a value of 16, which gives a faster training process. Adam optimizer (Kingma and Ba 2014), which generally has faster convergence with less memory compared to Adagrad, SGD (Ruder 2016) and RMSprop (Tieleman et al., 2012), has been selected. A dropout with a value of 0.5 gave satisfactory regularization and prevented overfitting of the model. Along with the proposed model, different competing models have also been experimented with the same settings of hyper-parameters. Selecting an appropriate loss function is essential in evaluating and enhancing the overall performance of deep convolutional neural networks (DCNNs), along with hyper-parameters. Because of the presence of thirty-nine classes in the dataset, the categorical cross-entropy loss (CCE loss) was selected as an ideal choice for the Softmax classifier over binary cross-entropy (Too et al. 2019). In Eq. (3), N denotes the thirty-nine classes, and for every diagnosed class p in each instance i of N, the model computed the count of errors made on each observation j based on its true values y. For each diagnosis p, the loss is computed using the natural log function (https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html).

$$\mathrm{CCE}_{\mathrm{loss}} = -\sum_{i}^{N} y_{j,i} \log\left(p_{j,i}\right) a \tag{3}$$

Performance metrics

This research study employed a confusion matrix to evaluate and visualize the interpretation of efficacy of the presented model in accurately detecting and categorizing plant diseases. The diagonal value in the confusion matrix represents the number of correct instances for a specific class. Considering the dataset contains more than two classes, the present study can be classified as a multi-class classification task. The performance metrics were computed using the indices described in Eqs. (4)–(7). True Positive (TP) is the count of instances in which the model accurately predicts a positive outcome in a specific category, indicating that both the prediction and the actual result are positive. True Negative (TN) gives the count of the instances where the model correctly predicts a negative outcome, indicating that both the prediction and the actual result are negative. False Positive (FN) is the count of the cases where the model predicts a negative outcome, but the actual result is positive, and False Positive (FP) is the count of the cases where the model predicts a positive result, but the actual result is negative (Hossin and Sulaiman 2015). Figure 2 displays the confusion matrix of the proposed model. For instance, 89 out of 100 samples have been correctly identified with 11 misclassifications in the Class 11, "Corn Northern Leaf Blight". For the remaining classes, the proposed method gives good results and differentiates all the diseased categories very well.

In this research, several performance metrics have been utilized to assess the efficacy of the proposed LWDN in accurately identifying and classifying plant diseases. Specifically, the metrics accuracy (Acc), precision (Prec), recall (Rec), and F1 score have been used. Accuracy is defined as the proportion of correctly classified cases out of the total cases in the dataset. It gives the ratio of correctly classified plant samples. Precision measures the proportion of correctly classified positive samples (i.e. disease-infected) among all the samples classified as positive. Recall measures the proportion of percentage of positive samples (i.e. disease-infected) correctly classified among all the actual positive samples in the dataset. The F1 score is computed as the harmonic mean of precision and recall, serving as a consolidated score for evaluating the overall performance of a classification system. It offers a comprehensive evaluation of the model's performance, considering both precision and recall simultaneously.

For x is disease category/class:

$$\mathrm{Prec}(x) = \frac{\mathrm{TP}(x)}{\mathrm{TP}(k) + \mathrm{FP}(k)} \tag{4}$$

Table 3 Settings of training hyper-parameter

Hyper-parameters	Value	Description
Learning rate	0.0001	Controls the weights of the model with respect to the loss gradient
Batch size	16	Number of images used in an iteration
Optimizer	Adam	To reduce the loss function during the training phase by adjusting the neural network parameters
Epochs	50	Complete iteration over a given dataset during the training phase
Dropout rate	0.5	To prevent overfitting

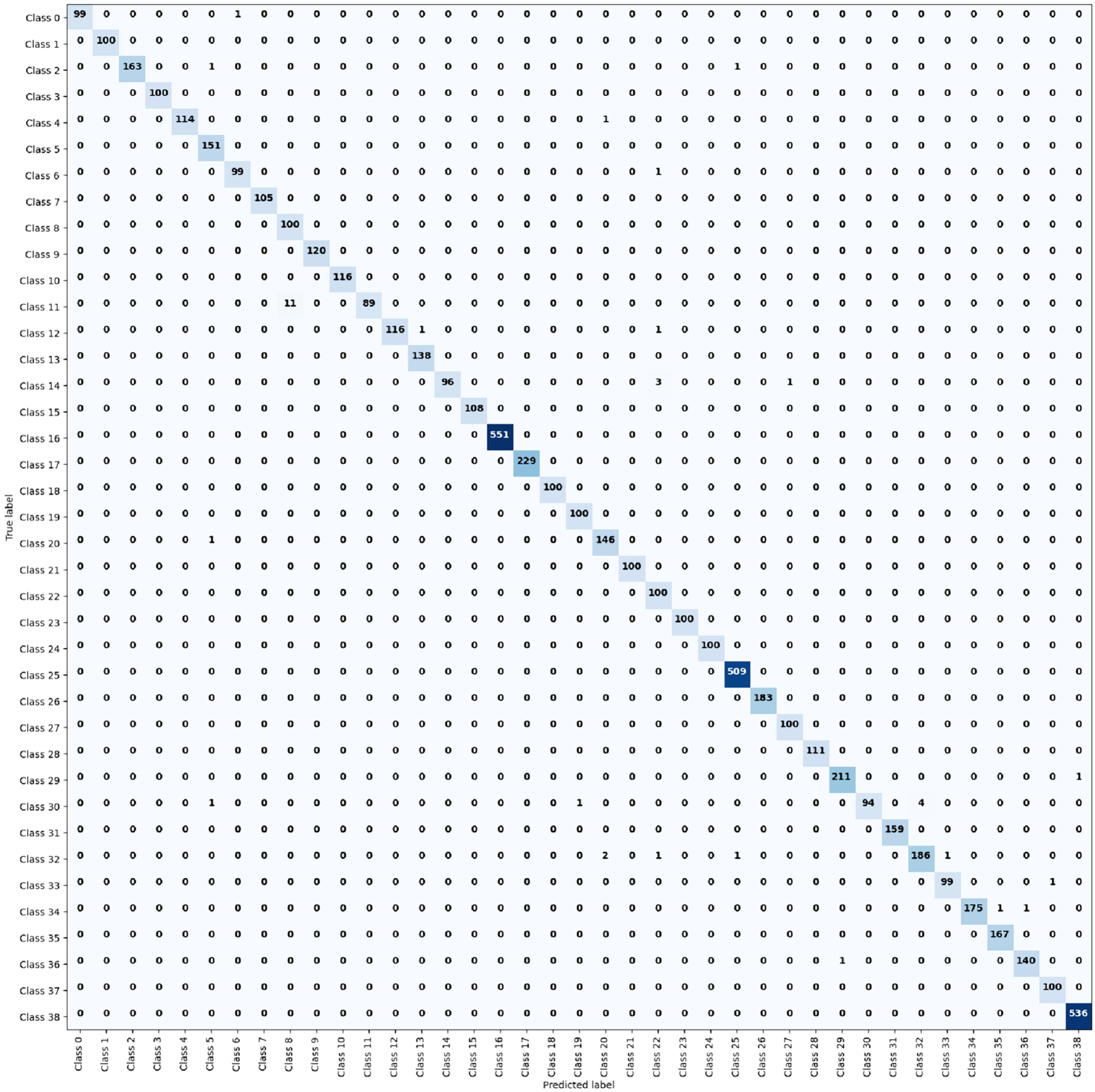

Fig. 2 LWDN model's confusion matrix

$$\mathrm{Rec}(x) = \frac{\mathrm{TP}(x)}{\mathrm{TP}(k) + \mathrm{FN}(k)} \tag{5}$$

$$F1\mathrm{Score}(x) = \frac{2 * \mathrm{Prec}(x) * \mathrm{Rec}(x)}{\mathrm{Prec}(x) + \mathrm{Rec}\ (x)} \tag{6}$$

$$\mathrm{Acc}\ (x) = \frac{\mathrm{TP}(x) + \mathrm{TN}(x)}{\mathrm{TP}(x) + \mathrm{TN}(x) + \mathrm{FP}(x) + \mathrm{FN}(x)} \tag{7}$$

In addition, the validated DCNNs were subjected to performance analysis using various data visualization techniques, such as learning curves, area under the receiver operating characteristic (AUROC), and area under the precision–recall (AUPR). Floating point operations or FLOPs, defined as number of operations required by model for the classification task, were calculated for all models using Keras-flops python package (Tokusumi 2020).

Experiment study and results

This section discusses the experimental findings in detail and a comparative study of various models.

Experiment setup

In this experimental study, the Python programming language was used, and all experiments were conducted on NVIDIA DGX GPU servers. These servers were equipped with 512 GB RAM and 8 high-speed Tesla V100 GPUs, with each GPU having a capacity of 32 GB. The deep learning package employed for the study was Keras, with TensorFlow serving as the backend. Pre-trained convolutional neural network (CNN) models from Keras applications, including VGG16, DenseNet121, MobileNetV2, NasNetMobile, Xception, EfficientNet B0, and InceptionV3, were considered in this study.

Dataset preparation

In this research, the dataset preparation involved several steps. First, the dataset was partitioned into three distinct subsets, namely training, testing and validation set, with respective proportions of 80%, 10%, and 10%. In the second step, all the images in the dataset were resized to dimensions of 224 × 224 pixels. The third step involved normalizing the image intensity values to reduce the network computational complexity. Specifically, the procedure of normalizing the intensity values of each pixel involved dividing them by 255, resulting in a normalized numerical range between 0 and 1.

Performance comparison with CNN models

In this research, the performance of the proposed work was analysed by comparing it with seven CNNs that were evaluated as benchmark methods. These CNNs included MobileNetV1, MobileNetV2, EfficientNetB0, NASNetMobile, DenseNet, and XceptionNet models and were established using transfer learning that used pre-trained weights from ImageNet. In these CNN models, the classification layer was eliminated, and a new fully connected Softmax layer was

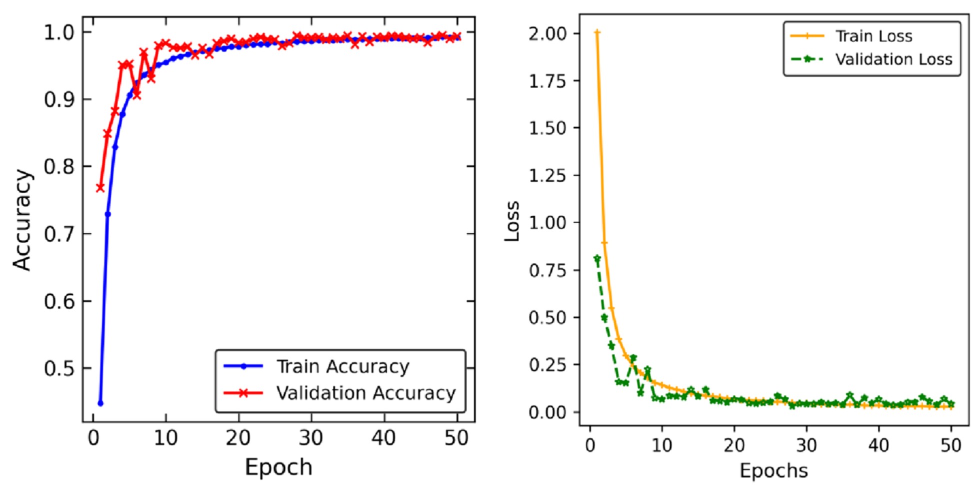

Fig. 3 Accuracy and loss curve of LWDN model

Table 4 Work comparison under various CNN models on PlantVillage dataset

Model name	Precision	Recall	F1 score	Accuracy	Total training time (min)	Time per epoch (min)	Total parameters	FLOPs (G)
DenseNet121	0.9966	0.9966	0.9966	0.9966	199.42	3.99	7,703,655	5.7
VGG16	0.9799	0.9784	0.9783	0.9784	174.63	3.49	15,118,695	30.7
Xception	0.9928	0.9901	0.9897	0.9901	341.87	6.84	22,051,919	9.14
EfficientNet B0	0.9970	0.9969	0.9969	0.9969	227.03	4.54	4,846,794	0.796
InceptionV3	0.9919	0.9917	0.9918	0.9917	153.30	3.07	22,993,223	5.7
MobileNetV2	0.9927	0.9925	0.9925	0.9925	159.03	3.18	3,055,207	0.614
NasNetMobile	0.9946	0.9945	0.9945	0.9945	412.38	8.25	4,952,251	1.15
LWDN (Proposed)	0.9939	0.9937	0.9936	0.9937	104.57	2.09	1,521,319	5.83

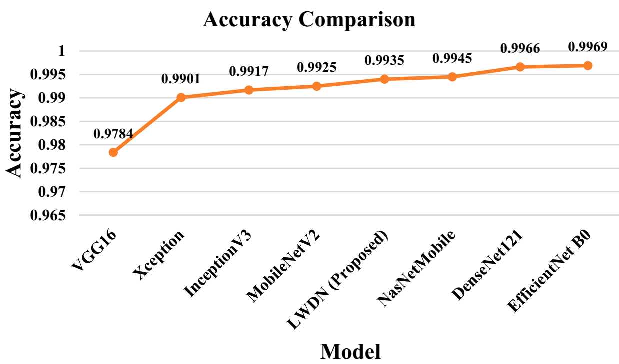

Fig. 4 Accuracy of different CNN models

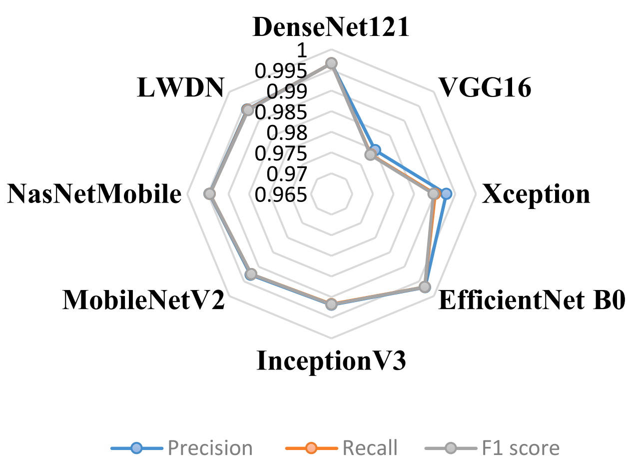

Fig. 5 Radar chart for CNN models on dataset

added with the actual number of classes in the dataset. The CNN models were then trained and subjected to extensive experimentation on the PlantVillage dataset. Figure 3 displays the accuracy and loss curve of the proposed method, while performance comparisons of various CNN models are presented in Table 4. The proposed method demonstrated acceptable performance during model training, with high accuracy and low loss, as shown in Fig. 3. The plots of both training and validation show similar trends, with similar values, as accuracy is continuously maximized, and loss is minimized. The accuracy score of all experimented models in ascending order is shown in Fig. 4. Performance comparison of proposed LWDN and other competing CNN models is visualized using a radar/spider chart and shown in Fig. 5, which uses three evaluation metrics. Model covering the maximum area in radar chart is better, thus LWDN is better than some model in terms of performance as shown in Fig. 5. Table 4 indicates that the proposed LWDN has a competitive performance of 99.37% which is comparatively higher than VGG16, MobileNetV2, XceptionNet, and InceptionV3. The proposed model takes the least time for training, with 104.57 min, compared to competing CNN models. LWDN has a total of 1,521,319 parameters, the least among the CNN model listed in Table 4. The LWDN model was able to obtain 99.39% precision and 99.37% recall value when experimented on PlantVillage. EfficientNet B0 achieved the highest performance due to the use of compound scaling method, with 99.69% accuracy with 4.8 M parameters among all models. DenseNet121, EfficientNet B0, and NasNetMobile have better performance with 99.66, 99.69, and 99.45 per cent accuracy, but these models have an extensive amount of parameters with 7.7 M, 4.8 M, and 4.9 M parameters and more training time. The proposed LWDN takes 93% fewer parameters compared to InceptionV3 and Xception, 90% fewer parameters compared to VGG16, and 50% fewer parameters compared to MobileNetV2. MobileNetV2 takes the least amount of computing power (FLOPs) due to the use of NAS technology but has a significantly higher parameter count and lower performance than LWDN. While FLOPs (floating point operations) is not the sole determinant of whether a model is lightweight, it is an important factor to consider alongside other metrics, such as number of parameter and training time. In the proposed model, floating point operations is 5.83G because we concatenated two pruned DenseNet models. However, training time and parameter amount are comparatively lower than other models, thus rendering it a lightweight model.

LWDN achieved good results with a performance score of over 99%. Some samples have been misclassified. For instance, 39 samples out of 6149 have been inaccurately identified. Average F1 score of the LWDN model yielded 99.36%.

In order to perform a comprehensive evaluation, this study employed AUROC analysis to visually capture the balance between the sensitivity and specificity of the models. A higher AUROC indicates better performance of a DCNN model, while an AUROC value of < 0.5 implies that the model cannot effectively distinguish or identify a particular case (Geetharamani and Pandian 2019). The proposed model demonstrated outstanding sensitivity and specificity performance, with a consistent AUC of 1.00 for all thirty-nine classes in the PlantVillage dataset.

In addition to the AUROC, the AUPR curve is also commonly used as a graphical metric for evaluating model performance, particularly in cases of unbalanced data distribution (Jeni et al. 2013). Unlike AUROC, AUPR emphasises on the count of incorrect diagnoses cases while still considering the region under the curve. Except for five classes, the proposed LWDN model attained a micro-average AUPR of 1.00 for the majority of categories in the PlantVillage dataset. However, the model's performance in class 4, class 8, class 11, class 20, class 29, class 30, class 32, and class 36 has an AUPR of 0.992, 0.987, 0.988, 0.998, 0.999, 0.992, 0.998, and 0.999, respectively, that indicate a challenge in accurately diagnosing these diseases. For the remaining classes, model have AUPR score of 1.

Statistical analysis of results

To validate the experimental results on PlantVillage, Friedman statistical test was utilized (Friedman 1937). The statistical test can determine whether the CNN models differ significantly from each other. The null hypothesis (H_0) assumed here is that all models have the same performance. The alternative hypothesis (H_1) assumed here is that there is at least one model that outperforms at least one other model. The rationale for selecting the Friedman test stems from its high statistical power when the number of compared entities exceeds five.

For a given D dataset and n models to be compared, models are ranked on a scale of 1 (worst) to n (best). The final rank of the models will be computed by averaging the rank over all the datasets. We have experimented on the PlantVillage dataset only, so the average of the rank is not needed. The rejection or acceptance of null and alternative hypothesis depends on p value test statistics. If the p value is lesser than 0.05, it indicates that there is a significant difference among all the models, and null hypothesis is rejected. Here, we considered precision, recall, and F1 score for Friedman statistic test. Table 5 shows the ranking of the models using Friedman test. The results indicate that the p value of 0.0049, obtained from the test, is less than 0.05, indicating that the null hypothesis of no significant difference between the models is rejected. Therefore, it can be inferred that considerable difference exists among all the models.

Model size comparison

In terms of computational time, LWDN outperforms the other seven CNN models. Additionally, LWDN has the smallest model file size of 13.8 MB, making it a suitable option for implementation on mobile devices that are constrained by limited storage capacity. The benefits of using LWDN include its ability to achieve competitive performance with a reduced number of parameters. EfficientNet B0, the best-performing model among all, has a model file size of 59 MB, which is 77% larger than LWDN. Figure 6 illustrates model file sizes comparison of the various CNN models.

Table 5 Ranking of CNN models using Friedman test on various performance metric on PlantVillage dataset

Model name	Rank		
	Precision	Recall	F1 score
DenseNet121	2	2	2
VGG16	3	2	1
Xception	3	2	1
EfficientNet B0	3	1.5	1.5
InceptionV3	3	1	2
MobileNetV2	3	1.5	1.5
NasNetMobile	3	1.5	1.5
LWDN (Proposed)	3	2	1
***p* value**	0.0049		

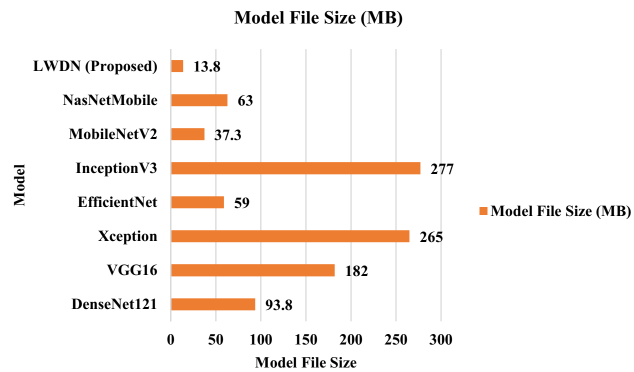

Fig. 6 File size of different experimented CNN model

Performance comparison against previous research

Table 6 illustrates the comparative performance analysis of the proposed study with some compact models developed. These studies used either of original or augmented PlantVillage dataset. The presented study has an accuracy of 99.37% with 1.5 M parameters and outperforms the work by Thakur et al. (2023), Kaya and Gürsoy (2023) and Arun and Umamaheswari (2023), which provides an accuracy of 99.16% with 6 M parameters, 98.17% with 8.13 M parameters and 98.14% with 2.87 M parameters, respectively. The proposed model is the smallest CNN model in terms of parameter size, developed for the whole PlantVillage dataset with remarkable performance, justifying the computational betterment of the proposed study.

Ablation studies

To showcase the efficiency of the LWDN architecture, an ablation study was conducted. This study involved removing certain parts of the deep learning architecture to evaluate their contributions to the overall network performance. The objective of the ablation analysis was to measure the robustness of the deep learning architecture's performance against structural changes caused by ablations, where layers and blocks were either added or removed. Specifically, one block was removed from the model at a time, and the model's performance was evaluated without the removed

Table 6 Computational comparison of the LWDN model with various prior research studies

Study	Method	Accuracy (%)	Number of parameters
Geetharamani and Pandian (2019)	Nine-layer CNN model	96.46	–
Mohanty et al. (2016)	GoogleNet	99.35	7 M
Too et al. (2019)	DenseNet121	99.75	7.1 M
Sutaji and Yıldız (2022)	Ensemble MobileNetV2 and Xception	99.10	26.5 M
Hanh et al. (2022)	EfficientNet B3 and EfficientNet B5	99.99 on both original and augmented PlantVillage dataset	10.6 M and 28.2 M for EfficientNet B3 and B5, respectively
Kaya and Gürsoy (2023)	Multi-headed DenseNet	98.17	8.13 M
Arun and Umamaheswari (2023)	PCCDL-PSCT	98.14	2.87 M
Thakur et al. (2023)	VGG-ICNN	99.16	6 M
LWDN (Proposed Study)	Modified DenseNet121 named lightweight DenseNet (LWDN)	99.37	1.5 M

block. Names of models resulting from these ablations are as follows.

LWDN_6_2: Lightweight DenseNet121 architecture with six blocks followed by a transition layer and two blocks left and replicated the same architecture and concatenated these two architectures.
LWDN_6_3: Lightweight DenseNet121 architecture with six blocks followed by a transition layer and three blocks left and replicated the same architecture and concatenated these two architectures.
LWDN_6_5: Lightweight DenseNet121 architecture with six blocks followed by transition layer and five blocks left and replicated the same architecture and concatenated these two architectures.
LWDN_1: Lightweight DenseNet121 architecture with six blocks followed by a transition layer and four blocks left.

Originally, the proposed LWDN model consisted of six dense blocks, succeeded by a transition layer, and followed by four dense blocks.

One by one, these dense blocks are removed, or one dense block is added to architecture and are named as mentioned above. The performance comparison of the ablated models is summarized in Table 7. All the models experimented with the same set of hyper-parameters. As the results show, removing or adding any block to the LWDN architecture worsens the performance of the model. Thus, the presented architecture has the optimum performance. LWDN has the best performance with 1.5 M parameters among all these ablated models. It should be noted that increasing the number of dense blocks in LWDN will lead to an increase in computational load, resulting in a slower training process, and there may not be a significant improvement in accuracy. Though the reduction in dense block decreases the computation complexity, performance is not improved. Among all the ablated models, LWDN have the best performance accuracy, while LWDN_1 takes the least number of parameters but has 98.65% accuracy. The performance of the LWDN_6_5, which has one more block than the proposed LWDN, is 99.29%. LWDN_6_5 takes 6.05G FLOPs and 9% more parameters amount than LWDN, thus justifying the optimum pruning of the existing DenseNet architecture. Therefore, it is relevant to infer that the proposed LWDN possesses a good trade-off among accuracy, parameter amount, and training time.

Conclusion

It is imperative to develop memory-efficient CNN models that can still achieve high levels of accuracy in image identification, with easy deployment on portable and mobile

Table 7 Performance of ablated models on the PlantVillage dataset

Model Name	Precision	Recall	F1 Score	Accuracy	Number of parameters	Training time (min)	FLOPs (G)
LWDN_6_5	0.9932	0.9929	0.9928	0.9929	1,680,039	109.27	6.05
LWDN_6_3	0.9880	0.9874	0.9873	0.9874	1,371,047	101.07	5.62
LWDN_6_2	0.9876	0.9865	0.9865	0.9865	1,229,223	97.3	5.43
LWDN_1	0.9874	0.9865	0.9866	0.9865	897,127	92.45	2.91
LWDN (Proposed)	0.9939	0.9937	0.9936	0.9937	1,521,319	104.57	5.83

devices. This investigation aimed to examine the capabilities of a lightweight and efficient network architecture with competitive performance results that can fulfil the necessary design specifications for embedded and mobile vision applications. In this research study, a lightweight deep learning model named lightweight DenseNet (LWDN) has been developed by pruning the DenseNet121 architecture for plant disease identification. We pruned the majority of layers of the original DenseNet121, replicated the model and then concatenated these models to generate the robust feature set. Pruning of the DenseNet121 model has been done in such a way that the overall parameter count is approximately 1.5 M, with no significant reduction in the performance of the model for plant disease diagnosis. DenseNet121 model was pruned and then replicated. These two pruned models were trained in different way to generate distinct feature set. In one of the pruned models, all layers were frozen, and in another one, all layers were unfrozen and trained on ImageNet and PlantVillage dataset. Additionally, this work also considered the implementation of seven other state-of-the-art models. For testing the performance of the model, the PlantVillage dataset has been used. The proposed LWDN has an advantage over other state-of-the-art CNN architectures in terms of fewer parameter sizes and lesser computational cost and complexity. LWDN is a pruned version of DenseNet121 architecture with six dense blocks followed by a transition layer and another four dense blocks. The trade-off between accuracy and computational cost has driven the development of a compact and computationally inexpensive LWDN model through the utilization of pruning and concatenation techniques. It was found that the training time of the proposed LWDN was minimal, with 104.57 min. With a lightweight design and a substantially reduced parameter size of 1.5 M, LWDN outperforms some of the comparatively larger models like MobileNetV2, VGG16, Xception, and InceptionV3 and attained a success rate of 99.37% on the PlantVillage dataset with 50% fewer parameters compared to MobileNetV2. By replicating the pruned network, LWDN has improved feature production. Thus, even with a small network structure, the proposed model reported remarkable accuracy towards plant disease identification. As evidenced by the results obtained in this study, the proposed technique of re-structuring and training the DCNN model like DenseNet121 can significantly preserve its performance while simultaneously saving on disc capacity and computing cost. The proposed lightweight model LWDN can be deployed on mobile and portable devices with limited computational capacity and storage. Furthermore, future researchers can use this method for other benchmark datasets and with a different model to produce realistic and better results. Additionally, further investigation can be carried out for low-performance results on some classes.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s41348-024-00915-z.

References

Abbas A, Jain S, Gour M, Vankudothu S (2021) Tomato plant disease detection using transfer learning with C-GAN synthetic images. Comput Electron Agric 187:106279

Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Van Esesn BC, Awwal AAS, Asari VK (2018) The history began from alexnet: a comprehensive survey on deep learning approaches. arXiv preprint arXiv:1803.01164.

Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L (2021) Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8:1–74

Arun RA, Umamaheswari S (2023) Effective multi-crop disease detection using pruned complete concatenated deep learning model. Expert Syst Appl 213:118905

Atila Ü, Uçar M, Akyol K, Uçar E (2021) Plant leaf disease classification using EfficientNet deep learning model. Eco Inform 61:101182

Basavaiah J, Arlene Anthony A (2020) Tomato leaf disease classification using multiple feature extraction techniques. Wireless Pers Commun 115(1):633–651

Bevers N, Sikora EJ, Hardy NB (2022) Soybean disease identification using original field images and transfer learning with convolutional neural networks. Comput Electron Agric 203:107449

Brahimi M, Boukhalfa K, Moussaoui A (2017) Deep learning for tomato diseases: classification and symptoms visualization. Appl Artif Intell 31(4):299–315

Cai Y, Zhang Z, Yan Q, Zhang D, Banu MJ (2021) Densely connected convolutional extreme learning machine for hyperspectral image classification. Neurocomputing 434:21–32

ML Cheatsheet, c2017. [Online]. Available: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html. Accessed 06 Sep 2023

Chen J, Chen J, Zhang D, Sun Y, Nanehkaran YA (2020a) Using deep transfer learning for image-based plant disease identification. Comput Electron Agric 173:105393

Chen J, Zhang D, Nanehkaran YA (2020b) Identifying plant diseases using deep transfer learning and enhanced lightweight network. Multimed Tools Appl 79:31497–31515

Chen J, Zhang D, Suzauddola M, Zeb A (2021a) Identifying crop diseases using attention embedded MobileNet-V2 model. Appl Soft Comput 113:107901

Chen J, Zhang D, Zeb A, Nanehkaran YA (2021b) Identification of rice plant diseases using lightweight attention networks. Expert Syst Appl 169:114514

Chuanlei Z, Shanwen Z, Jucheng Y, Yancui S, Jia C (2017) Apple leaf disease identification using genetic algorithm and correlation based feature selection method. Int J Agric Biol Eng 10(2):74–83

Dahl GE, Sainath TN, Hinton GE (2013) Improving deep neural networks for LVCSR using rectified linear units and dropout. In: 2013 IEEE international conference on acoustics, speech and signal processing, IEEE. pp 8609–8613

Das D, Santosh KC, Pal U (2020) Truncated inception net: COVID-19 outbreak screening using chest X-rays. Phys Eng Sci Med 43:915–925

Dheeraj A, Chand S (2023) Deep learning model for automated image based plant disease classification. In: Proceedings of international

conference on intelligent vision and computing (ICIVC 2022), Vol. 1. Springer Nature Switzerland. Cham, pp 21–32
Dubey SR, Jalal AS (2016) Apple disease classification using color, texture and shape features from images. SIViP 10:819–826
Fan X, Luo P, Mu Y, Zhou R, Tjahjadi T, Ren Y (2022) Leaf image based plant disease identification using transfer learning and feature fusion. Comput Electron Agric 196:106892
Fang S, Wang Y, Zhou G, Chen A, Cai W, Wang Q, Hu Y, Li L (2022) Multi-channel feature fusion networks with hard coordinate attention mechanism for maize disease identification under complex backgrounds. Comput Electron Agric 203:107486
Ferentinos KP (2018) Deep learning models for plant disease detection and diagnosis. Comput Electron Agric 145:311–318
Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701
Fu L, Li S, Sun Y, Mu Y, Hu T, Gong H (2022) Lightweight-CNN for apple leaf disease identification. Front Plant Sci, 1508.
Geetharamani G, Pandian A (2019) Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Comput Electr Eng 76:323–338
Gokulnath BV (2021) Identifying and classifying plant disease using resilient LF-CNN. Eco Inform 63:101283
Hanh BT, Van Manh H, Nguyen NV (2022) Enhancing the performance of transferred efficientnet models in leaf image-based plant disease classification. J Plant Dis Prot 129(3):623–634
He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770–778
Hossin M, Sulaiman MN (2015) A review on evaluation metrics for data classification evaluations. Int J Data Min Knowledge Manag Process 5(2):1
Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4700–4708
Isikdogan LF, Nayak BV, Chyuan-Tyng W, Moreira JP, Rao S, Michael G (2020) Semifreddonets: Partially frozen neural networks for efficient computer vision systems. In: Vedaldi A, Bischof H, Brox T, Frahm J-M (eds) Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII. Springer International Publishing, Cham, pp 193–208. https://doi.org/10.1007/978-3-030-58583-9_12
Jackulin C, Murugavalli S (2022) A comprehensive review on detection of plant disease using machine learning and deep learning approaches. Measur Sens 24:100441
Jeni LA, Cohn JF, De La Torre F (2013) Facing imbalanced data--recommendations for the use of performance metrics. In: 2013 Humaine association conference on affective computing and intelligent interaction. IEEE. pp 245–251
Jiang F, Lu Y, Chen Y, Cai D, Li G (2020) Image recognition of four rice leaf diseases based on deep learning and support vector machine. Comput Electron Agric 179:105824
Joshi RC, Kaushik M, Dutta MK, Srivastava A, Choudhary N (2021) VirLeafNet: automatic analysis and viral disease diagnosis using deep-learning in Vigna mungo plant. Eco Inform 61:101197
Kamal KC, Yin Z, Wu M, Wu Z (2019) Depthwise separable convolution architectures for plant disease classification. Comput Electron Agric 165:104948
Karthik R, Alfred JJ, Kennedy JJ (2023) Inception-based global context attention network for the classification of coffee leaf diseases. Eco Inform 77:102213
Kaur P, Pannu HS, Malhi AK (2019) Plant disease recognition using fractional-order Zernike moments and SVM classifier. Neural Comput Appl 31:8749–8768
Kaya Y, GÜrsoy E (2023) A novel multi-head CNN design to identify plant diseases using the fusion of RGB images. Ecol Inform 75:101998
Kılıç C, İnner B (2022) A novel method for non-invasive detection of aflatoxin contaminated dried figs with deep transfer learning approach. Eco Inform 70:101728
Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
Kumar S, Sharma B, Sharma VK, Sharma H, Bansal JC (2018) Plant leaf disease identification using exponential spider monkey optimization. Sustain Comput Inf Syst 28:100283
Kurmi Y, Gangwar S, Agrawal D, Kumar S, Srivastava HS (2021) Leaf image analysis-based crop diseases classification. SIViP 15(3):589–597
Liu B, Zhang Y, He D, Li Y (2017) Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry 10(1):11
Liu G, Peng J, El-Latif AAA (2023) SK-MobileNet: a lightweight adaptive network based on complex deep transfer learning for plant disease recognition. Arab J Sci Eng 48(2):1661–1675
Ma J, Du K, Zheng F, Zhang L, Gong Z, Sun Z (2018) A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. Comput Electron Agric 154:18–24
Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. Front Plant Sci 7:1419
Montalbo FJP (2021) Diagnosing Covid-19 chest x-rays with a lightweight truncated DenseNet with partial layer freezing and feature fusion. Biomed Signal Process Control 68:102583
Montalbo FJP (2022) Diagnosing gastrointestinal diseases from endoscopy images through a multi-fused CNN with auxiliary layers, alpha dropouts, and a fusion residual block. Biomed Signal Process Control 76:103683
Mustafa MS, Husin Z, Tan WK, Mavi MF, Farook RSM (2020) Development of automated hybrid intelligent system for herbs plant classification and early herbs plant disease detection. Neural Comput Appl 32:11419–11441
Naik BN, Malmathanraj R, Palanisamy P (2022) Detection and classification of chilli leaf disease using a squeeze-and-excitation-based CNN model. Eco Inform 69:101663
Nigam S, Jain R, Marwaha S, Arora A, Haque MA, Dheeraj A, Singh VK (2023) Deep transfer learning model for disease identification in wheat crop. Eco Inform 75:102068
Pandey A, Jain K (2022) A robust deep attention dense convolutional neural network for plant leaf disease identification and classification from smart phone captured real world images. Eco Inform 70:101725
Pantazi XE, Moshou D, Tamouridou AA (2019) Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers. Comput Electron Agric 156:96–104
Prajapati HB, Shah JP, Dabhi VK (2017) Detection and classification of rice plant diseases. Intell Decis Technol 11(3):357–373
Ramcharan A, Baranowski K, McCloskey P, Ahmed B, Legg J, Hughes DP (2017) Deep learning for image-based cassava disease detection. Front Plant Sci 8:1852
Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.
Sachar S, Kumar A (2021) Survey of feature extraction and classification techniques to identify plant through leaves. Expert Syst Appl 167:114181
Savary S, Willocquet L, Pethybridge SJ, Esker P, McRoberts N, Nelson A (2019) The global burden of pathogens and pests on major food crops. Nat Ecol Evolut 3(3):430–439
Sharma V, Tripathi AK, Mittal H (2023) DLMC-Net: Deeper lightweight multi-class classification model for plant leaf disease detection. Eco Inform 75:102025

Shin J, Chang YK, Heung B, Nguyen-Quang T, Price GW, Al-Mallahi A (2021) A deep learning approach for RGB image-based powdery mildew disease detection on strawberry leaves. Comput Electron Agric 183:106042

Shoaib M, Shah B, Ei-Sappagh S, Ali A, Ullah A, Alenezi F, Gechev T, Hussain T, Ali F (2023) An advanced deep learning models-based plant disease detection: a review of recent research. Front Plant Sci 14:1158933

Shrivastava VK, Pradhan MK (2021) Rice plant disease classification using color features: a machine learning paradigm. J Plant Pathol 103:17–26

Singh UP, Chouhan SS, Jain S, Jain S (2019) Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease. IEEE Access 7:43721–43729

Sutaji D, Yıldız O (2022) LEMOXINET: Lite ensemble MobileNetV2 and Xception models to predict plant disease. Eco Inform 70:101698

Thakur PS, Khanna P, Sheorey T, Ojha A (2022) Trends in vision-based machine learning techniques for plant disease identification: a systematic review. Expert Syst Appl 208:118117

Thakur PS, Sheorey T, Ojha A (2023) VGG-ICNN: a Lightweight CNN model for crop disease identification. Multimed Tools Appl 82(1):497–520

Tieleman T, Hinton G (2012) Rmsprop: divide the gradient by a running average of its recent magnitude. Coursera: neural networks for machine learning. COURSERA Neural Networks Mach. Learn, 17

Tiwari V, Joshi RC, Dutta MK (2021) Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images. Eco Inform 63:101289

Tokusumi (2020) Keras-flops calculator.

Too EC, Yujian L, Njuki S, Yingchun L (2019) A comparative study of fine-tuning deep learning models for plant disease identification. Comput Electron Agric 161:272–279

Turkoglu M, Yanikoğlu B, Hanbay D (2022) PlantDiseaseNet: Convolutional neural network ensemble for plant disease and pest detection. SIViP 16(2):301–309

Udendhran R, Balamurugan M (2021) Towards secure deep learning architecture for smart farming-based applications. Complex Intell Syst 7:659–666

Uğuz S, Uysal N (2021) Classification of olive leaf diseases using deep convolutional neural networks. Neural Comput Appl 33(9):4133–4149

Xiao Z, Shi Y, Zhu G, Xiong J, Jianhua W (2023) Leaf disease detection based on lightweight deep residual network and attention mechanism. IEEE Access 11:48248–48258. https://doi.org/10.1109/ACCESS.2023.3272985

Yang D, Wang F, Hu Y, Lan Y, Deng X (2021) Citrus huanglongbing detection based on multi-modal feature fusion learning. Front Plant Sci 12:809506

Yu M, Ma X, Guan H (2023) Recognition method of soybean leaf diseases using residual neural network based on transfer learning. Eco Inform 76:102096

Yu T, Zhu H (2020) Hyper-parameter optimization: a review of algorithms and applications. arXiv preprint arXiv:2003.05689.

Zhang S, Wu X, You Z, Zhang L (2017) Leaf image based cucumber disease recognition using sparse representation classification. Comput Electron Agric 134:135–141

Zhang K, Guo Y, Wang X, Yuan J, Ding Q (2019) Multiple feature reweight densenet for image classification. IEEE Access 7:9872–9880

Zhang Z, Flores P, Friskop A, Liu Z, Igathinathane C, Han X, Kim HJ, Jahan N, Mathew J, Shreya S (2022) Enhancing wheat disease diagnosis in a greenhouse using image deep features and parallel feature fusion. Front Plant Sci 13:834447