


Article

PDD-Net: Plant Disease Diagnoses Using Multilevel and Multiscale Convolutional Neural Network Features

Hamed Alghamdi and Turki Turki * 

Department of Computer Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia;
hhalghamdi0002@stu.kau.edu.sa

* Correspondence: tturki@kau.edu.sa

Abstract: Overlooked diseases in agriculture severely impact crop growth, which results in significant losses for farmers. Unfortunately, manual field visits for plant disease diagnosis (PDD) are costly and time consuming. Although various methods of PDD have been proposed, many challenges have yet to be investigated, such as early stage leaf disease diagnosis, class variations in diseases, cluttered backgrounds, and computational complexity of the diagnosis system. In this paper, we propose a Convolutional Neural Network (CNN)-based PDD framework (i.e., PDD-Net), which employs data augmentation techniques and incorporates multilevel and multiscale features to create a class and scale-invariant architecture. The Flatten-T Swish (FTS) activation function is utilized to prevent gradient vanishing and exploding problems, while the focal loss function is used to mitigate the impact of class imbalance during PDD-Net training. The PDD-Net method outperforms baseline models, achieving an average precision of 92.06%, average recall of 92.71%, average F1 score of 92.36%, and accuracy of 93.79% on the PlantVillage dataset. It also achieves an average precision of 86.41%, average recall of 85.77%, average F1 score of 86.02%, and accuracy of 86.98% on the cassava leaf disease dataset. These results demonstrate the efficiency and robustness of PDD-Net in plant disease diagnosis.

Keywords: plant disease diagnosis; computer vision; convolutional neural network; multilevel features; multiscale features; leaf diseases diagnosis; disease classification



Citation: Alghamdi, H.; Turki, T. PDD-Net: Plant Disease Diagnoses Using Multilevel and Multiscale Convolutional Neural Network Features. *Agriculture* **2023**, *13*, 1072. <https://doi.org/10.3390/agriculture13051072>

Academic Editor: Maciej Zaborowicz

Received: 11 March 2023

Revised: 12 May 2023

Accepted: 15 May 2023

Published: 17 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Agriculture is the backbone of the food supply chain [1], and the economies of numerous developing countries depend on it [2]. Outbreaks of plant diseases impact the condition of crops and reduce their yield, resulting in significant losses [3]. The loss of yields can influence the food supply chain and economy of the country. Plant infections are generally caused by bacteria, fungi, viruses, parasites, or environmental factors, and many farmers are unaware of some types of plant disease [4]. Early stage plant disease recognition helps control the disease and prevent widespread damage to crops [5]. To control plant disease outbreaks, regular and continuous consultations with experts are required. However, the regular visual inspection of experts in remote areas of developing countries is expensive, less accurate, and time consuming [6]. The automatic identification of crop disease symptoms is a very beneficial, fast, and cost-effective solution [7].

Generally, a crop's disease symptoms appear on the leaves. Additionally, digital images of leaves are strong candidates for plant disease recognition. During the last few years, plenty of expert systems have been proposed for crop disease diagnosis. These systems are grouped into two categories: (i) conventional computer vision-based diagnosis [7] and (ii) deep-learning-based [5,6] diagnosis. Conventional computer vision-based crop disease diagnosis systems are based on feature extraction and feature classification mechanisms [7]. In recent decades, well-known feature extraction approaches such as Local Binary Patterns (LBP) [8], Scale Invariant Features Transform (SIFT) [9], Supported Up Robust Features

(SURF) [10,11], Histogram of Oriented Gradients (HOG) [12], Gabor Transform (GT) [13], etc., have been used. Meanwhile, machine-learning (ML)-based classifiers such as Support Vector Machine (SVM) [11], Naive Bayes (NB) [14], Random Forest (RF) [14], Fisher Linear Discriminator (FLD) [15], etc., have been applied to classify leaf diseases. The accuracy of conventional diagnosis systems is highly dependent on hand-crafted feature extraction mechanisms and classification algorithms [16]. In particular, the selection of optimal and robust handcrafted features for disease identification is very challenging because every feature has its own limitations (i.e., scale variation, illumination variation, interclass variation, deformation variation, etc.) [17].

Recently, with the advent of domain-specific architectures [18] (i.e., GPU, TPU, etc.), deep learning (DL)-based classifiers, particularly CNN, are becoming increasingly popular [19–33]. A CNN-based classifier (i.e., VGG-Net [34], ImageNet [35], Inception-Net [36], DenseNet-121 [37], MobileNet [38], ResNet50 [39], etc.) automatically extracts and learns optimal low-level features for classification. Additionally, CNN-based architectures have shown good performance during image classification tasks [19]. Despite the promising results achieved by general CNN-based image classifiers, real-time plant-disease-classification applications face several challenges, such as early stage diagnosis [23,24], disease class variation [22,25], limited samples in benchmark datasets [24,29], and class imbalance problems of datasets [25]. Moreover, General CNN-based classification approaches often suffer from high computational costs and have many parameters to train [23].

To resolve these challenges, this research proposes a hybrid CNN architecture that combines the advantages of current advancements. The proposed model incorporates a spatial pyramid pooling block [40] that enhances the extraction of local and global multi-scale features, which improves the accuracy of small-scale lesion classification. A dense connection block [37] is introduced to maximize information flow within network layers and boost multilevel feature extraction capabilities. To decrease the model complexity and total training parameters, a global averaging layer [23] is introduced to replace the fully connected layers. The Flatten-T Swish (FTS) activation function [41] is used to overcome the gradient vanishing problem during backpropagation, which is commonly encountered with traditional activation functions. The FTS activation function provides a high convergence rate during training, which ensures a better classification performance. The proposed model is evaluated using the PlantVillage and cassava leaf disease datasets, and the results demonstrated significant improvements in the classification accuracy compared to traditional CNN-based methods.

In this study, the following significant contributions are made to address the limitations of existing computer vision-based systems for plant disease diagnosis (PDD):

- A novel CNN-based framework, PDD-Net, is proposed; it incorporates data augmentation techniques and transfer learning to efficiently handle limited data samples and enhance the model-training process.
- The PDD-Net architecture was designed to include multilevel and multiscale features, a stable FTS activation function, and a Global Average Pooling (GAP) layer, which together improve the classification accuracy and prevent overfitting in the model.
- To ensure model generalization and stability, fivefold cross-validation was employed as an essential part of the model evaluation process.
- The reported performance results such as precision, recall, F1 score, and accuracy demonstrated the superior performance of PDD-Net compared to baseline methods, including DenseNet-201, DenseNet-121, ResNet-50, and VGG-16.

Overall, this research contributes to the development of a robust and efficient PDD system that can handle early stage leaf disease diagnosis and class variation in plant diseases.

The remainder of this paper is organized as follows: Section 2 presents a review of the related work in PDD. Section 3 describes the materials and methods employed in our study, including the proposed PDD-Net framework. Section 4 provides a detailed analysis of the results obtained from our experiments. Section 5 contains the discussion. Finally, Section 6 concludes the paper and outlines potential future research directions.

2. Literature Review

In the field of artificial intelligence, two types of vision-based frameworks based on deep learning (DL) and machine learning (ML) are used for plant disease recognition. Although traditional ML-based models are lighter than sophisticated DL-based frameworks, their recognition accuracy is less promising than that of DL based models. As a result, this study concentrated solely on DL-based plant disease recognition frameworks.

For plant disease recognition, several Convolutional Neural Network (CNN)-based strategies have been suggested with promising results. However, these DL systems demand a huge number of training instances. Furthermore, the labeling of training data demands a comprehensive biological understanding.

To address such challenges, Cap et al. [42] introduced LeafGAN, a Generative Adversarial Network (GAN)-based approach used to augment training samples in plant disease classification tasks. This innovative augmentation technique employs image-to-image translation to generate novel data samples that enrich the training set and thereby potentially upgrade the overall performance and generalization of the classification model. In [29], Abbas et al. utilized DenseNet-121 [37] for plant disease classification. In their proposed framework, a Conditional GAN was used to enhance the training samples. Multilevel features are used to cope with class variations between plant diseases. Their proposed framework only targeted tomato plant diseases and did not consider the multiscale features crucial for early stage lesion recognition. In Ref. [24], the Dilated and Inception CNN (DICNN) paradigm for classifying cucumber diseases is explained. Two types of data augmentation, such as translation and rotation, have been used to improve training data in their architecture. In another study, Zhang et al. [20] utilized a thirteen-layer deep CNN architecture for fruit classification in supermarkets, factories, and other fields. They introduced three different data augmentation methods to minimize the learning data starvation problem of CNN models. They also observed that data augmentation enhanced the accuracy of the CNN-based classifier. In Ref. [31], Mostafa et al. designed a PDD framework for the guava plant. The image rotation technique was used for data augmentation. Their framework utilized unsharp masking and color-histogram equalization techniques along with deep learning features for disease identification.

Other than limited training data samples, plant disease diagnosis has faced several major challenges, including early stage detection, handling leaf disease class variation, and addressing dataset class imbalance. Researchers have proposed various techniques to address these issues by enhancing and modifying existing frameworks. They have proposed hybrid CNN architectures that combine various methodologies and techniques, aiming to mitigate the limitations of traditional classifiers. Wang et al. [26] designed an architecture that classifies and localizes plant diseases using object detection algorithms (i.e., Faster RCNN [43], YOLO [44], SSD [45], etc.) from digital images. They introduced the deep block attention to improve the feature extraction capability of their backbone architecture (i.e., VGG-16). Alatawi et al. [32] also utilized a VGG-16 [34] backbone architecture for plant disease classification. They utilized SoftMax and ReLu activation functions along with “Sparse Categorical Cross Entropy Loss” for CNN model training. To validate the performance of their architecture, they trained VGG-16 on nineteen selected classes of the PlantVillage benchmark, achieving a 95% testing accuracy for these selected classes. In [21], Chen et al. designed a CNN-based PDD framework by utilizing the VGG-19 [34] architecture. To enhance the VGG-19 model, they incorporated the inception module, which resulted in an improved performance. In [28], Wagle et al. suggested a compact CNN-based framework for leaf infection classification. Their compact CNN-based architecture utilized “Rectified Linear Unit” ReLu [46] activation functions throughout the network and fully connected layer before making a final prediction, which increased the total training parameters and overall model complexity. In [23], Zhang et al. suggested a deep-learning-based model called Global Pooling Dilated CNN (GPDCNN) for leaf disease classification. Their proposed CNN architecture is based on AlexNet architecture. To decrease the complexity of the CNN framework, the GPDCNN methodology uses a global

averaging layer [23] rather than fully connected layers (FCL). Furthermore, GPDCNN uses a dilated-CNN layer to efficiently recover spatial information from images, which is vital for small lesion detection. These CNN models ignore multilevel features that help diminish the influence of class variations among different plant diseases. Moreover, in their datasets, certain diseases appear with notably low frequency; the imbalanced nature of these datasets negatively impacted the model's training and overall performance.

To resolve the class imbalance, Zhong et al. [25] applied focal loss as an alternative to cross-entropy loss to rectify the imbalanced composition of plant disease datasets. Additionally, to cope with the class variation, the DenseNet-121 [37] architecture was used in their proposed multiclass classification scheme. Tiwari et al., 2021 [22] utilized DenseNet-201 for plant disease classification. To improve the DenseNet-201 basic design, they added four layers of dense blocks. Additionally, fivefold cross validation was employed to assess the model. Although their suggested CNN frameworks [22,25] are resilient to class diversity in plant diseases, these models failed to detect plant infections at an early stage. Tariq et al. [30] suggested a hybrid framework for plant infection classification. They utilized the ResNet-50 [39] deep learning model for training. In their framework, handcrafted features (i.e., color and LBP [8]) are fused with deep learning features. Research shows that feature fusion strategies enhance the robustness of monotonic gray-scale variations. However, the skip connection of ResNet-50 architecture increases the overall complexity of the model. In [27], Chakraborty et al. utilized CNN-based image classifiers (i.e., VGG-16 [34], VGG-19 [34], MobileNet [38], and ResNet50 [39]) for potato leaf disease identification. In [33], Eunice et al. utilized four (i.e., VGG-16 [34], DenseNet121 [37], Inception V4 [47], and ResNet-50 [39]) state-of-the-art CNN-based frameworks for plant infection classification. To handle the data nonlinearity, they utilized the ReLU activation function in CNN architectures. In [48], Attallah proposed a framework for tomato leaf infection classification using three different CNNs: ResNet-18, ShuffleNet, and MobileNet. The study employed transfer learning to extract features from the last fully connected layer, resulting in an additional high-level interpretation. Subsequently, features from all three CNNs were merged to harness the benefits of each structure. A hybrid feature selection approach was then applied to generate a comprehensive, lower-dimensional feature set. Attallah's study achieved notable progress in tomato leaf disease classification, but the pipeline's generalizability to other economically significant crops remains untested. A more versatile and broadly applicable pipeline could be achieved through the generalization and exploration of data augmentation methods to increase the model's robustness and performance, especially in cases with limited or imbalanced training data.

The literature highlights several research gaps in plant disease recognition using deep learning frameworks, including early stage detection, handling class variation, addressing dataset class imbalances, overcoming limited training data, managing model complexity, and improving scalability and generalizability. Addressing these gaps will provide precise and robust plant disease recognition models, which will ultimately benefit agriculture and food security. Therefore, we propose the PDD-Net architecture to overcome these challenges.

3. Materials and Methods

3.1. Image Preprocessing and Augmentation

To ensure the input parameters meet the needs of the CNN model, we preprocessed the dataset's images before feeding them to the CNN architectures. During preprocessing, each input image was resized to 224×224 dimensions. Then, normalization (i.e., image/255.0) was applied to ensure that all the data were described under the same distribution, which enhances the training convergence and stability [49]. Figure 1 displays the original dataset image alongside its preprocessed form, where it has been resized and normalized.

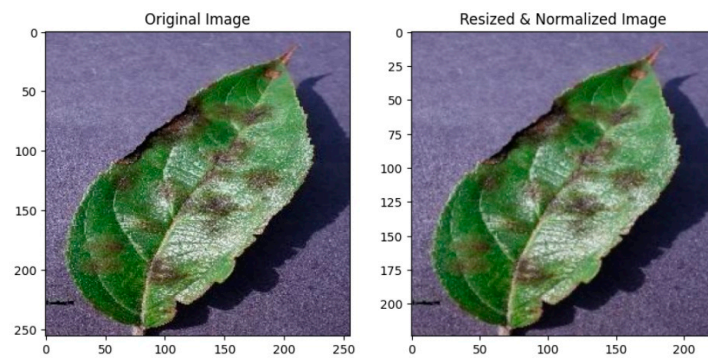


Figure 1. Image resizing and normalization applied to an image sample from PlantVillage benchmark dataset.

The CNN-based models' performance is strongly dependent on the size of the training samples [28]. To avoid overfitting (i.e., when a model discovers a function with particularly high variance to accurately characterize the training data) of the CNN architecture, large numbers of training examples are essential. To increase the training samples, three geometric transformation-based (i.e., not label-preserving transformation) augmentation methods (i.e., rotation, flipping, and noise injection) are used in this research, and some augmented samples are shown in Figure 2.

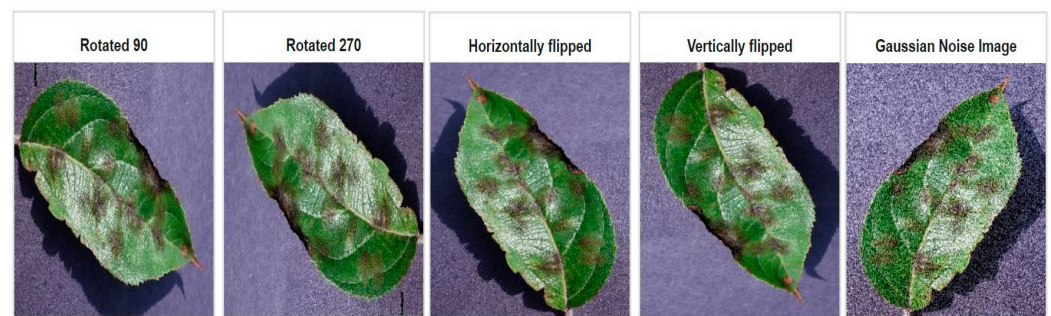


Figure 2. Image augmentation samples generated using a sample image of PlantVillage benchmark dataset.

3.2. Transfer Learning

Transfer learning is applied when information gained in one area is utilized in another [21,29]. Deep CNNs, which are used in deep learning, require a lot of training material. When the required volume of data is insufficient, pretrained CNN models are used for training. These networks have been previously trained on a significant number of images; the process of sharing existing knowledge to learn a new domain is known as transfer learning. A CNN's pretrained network is then fine-tuned using a relatively small quantity of new data to facilitate the transfer of knowledge. In this study, similar to other instances of image categorization, the accumulated knowledge is introduced by a pre-trained framework from the ImageNet task. ImageNet is a database that contains more than fourteen million images corresponding to 1000 standard classes.

3.3. Benchmark Acquisition

Two publicly available benchmarks known as cassava leaf disease (CLD) and PlantVillage are utilized to authenticate the performance of the designed PDD-Net.

The cassava leaf disease (CLD) benchmark dataset is employed to assess the functioning of the suggested PDD-Net architecture in this study. The dataset contains 21,397 annotated images of cassava plant leaves. These annotated images are classified into five categories based on their respective labels: (i) Healthy Cassava Leaves (HCL), (ii) Cassava Bacterial Blight (CBB), (iii) Cassava Brown Streak Disease (CBSD), (iv) Cassava Mosaic Disease (CMD), and (v) Cassava Green Mottle (CGM). In the CLD benchmark, the CMD class has significantly

more samples than all other classes. To increase the number of data samples and mitigate the impact of class imbalance, data augmentation methods are applied to all classes except the CMD category. After augmentation, the total number of data samples expanded to 54,353. A selection of data samples from the CLD benchmark dataset along the class labels are shown in Figure 3.

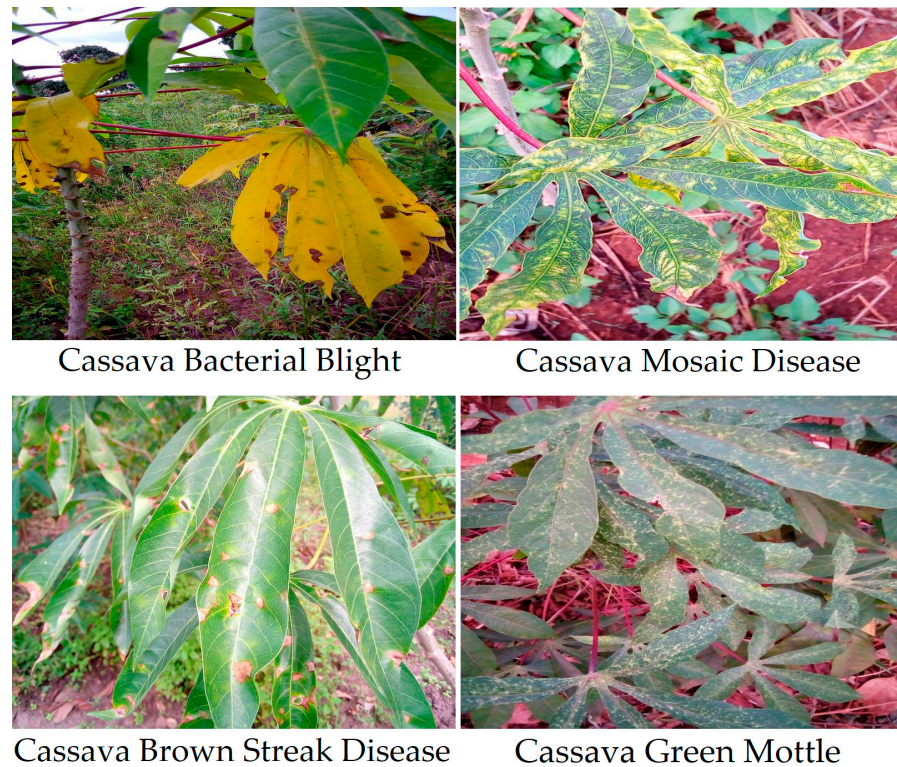


Figure 3. Some image samples from the cassava leaf disease (CLD) benchmark dataset.

To evaluate the PDD-Net architecture, the CLD benchmark dataset was partitioned into training and testing subsets, with 80% of the data allocated for training and the remaining 20% for testing. Table 1 presents a comprehensive overview of the division of images for every class within the training and test subsets.

Table 1. Summary of the cassava leaf disease (CLD) benchmark dataset employed in this study.

Category	Training Samples	Testing Samples	Total Samples
HCL	10,308	2577	12,885
CBB	4348	1087	5435
CBSD	8756	2189	10,945
CMD	10,526	2632	13,158
CGM	9544	2386	11,930
Total Samples	43,482	10,871	54,353

By distributing the CLD dataset into training and test subsets, this research aims to provide an accurate assessment of the PDD-Net architecture's performance in addressing the complex task of PDD.

The PlantVillage dataset contains 54,305 labeled pictures of plant leaves. It covers fourteen different types of plants, including (i) apple, (ii) blueberry, (iii) cherry, (iv) corn, (v) grape, (vi) orange, (vii) peach, (viii) pepper, (ix) potato, (x) raspberry, (xi) soyabean, (xii) squash, (xiii) strawberry, and (xiv) tomato. There are 38 different types of classification labels organized according to disease types. Some categories (i.e., cedar apple rust, peach healthy, grape healthy, potato healthy, raspberry healthy, strawberry healthy, and tomato mosaic virus) have very limited (i.e., less than 500) image samples. To overcome the data

limitation, different data augmentation techniques were utilized to artificially improve the data samples. Following data augmentation, the number of samples increased from 54,305 to 63,945. Following data augmentation, the benchmark images were randomly divided into an 80:20 train and test set. Table 2 reveals the details of the PlantVillage benchmark after applying various data augmentation techniques.

Table 2. The PlantVillage benchmark dataset after data augmentation.

Plant Name	Leaf Label	Image Frequency		Plant Name	Leaf Label	Image Frequency	
		Train	Test			Train	Test
Apple	Scab (AS)	504	126	Peeper	Bacterial spot (BS)	798	199
	Black rot (ABR)	497	124		Healthy (H)	1182	296
	Cedar apple rust (ACAR)	1100	275	Blueberry	Healthy (BH)	1202	300
	Healthy (AH)	1316	329	Orange	Huanglongbing (OH)	4406	1101
Cherry	Powdery mildew (CPM)	842	210	Raspberry	Healthy (RH)	1484	371
	Healthy (CH)	683	171	Soybean	Healthy (SH)	4072	1018
Corn/Maze	Gray leaf spot (MGLS)	410	103	Squash	Powdery mildew (SPM)	1468	367
	Common rust (MCR)	954	238	Strawberry	Leaf scorch (SLS)	887	222
	Northern leaf blight (MNLB)	788	197		Healthy (St-H)	1824	456
	Healthy (MH)	930	232	Tomato	Bacterial spot (TBS)	1702	425
Black rot (GBR)	944	236	Early blight (TEB)		800	200	
Black measles (GBM)	1106	277	Late blight (TLB)		1527	382	
Leaf blight (GLB)	861	215	Leaf mold (TLM)		762	190	
Healthy (GH)	1716	429	Septoria leaf spot (TSLS)		1417	354	
Peach	Bacterial spot (PBS)	1338	459		Spider mites (TSM)	1341	335
	Healthy (PH)	1440	360		Target spot (TTS)	1123	281
Potato	Early blight (Po-EB)	800	200	Mosaic virus (TMV)	1468	367	
	Late blight (Po-LB)	800	200	Yellow leaf curl virus (TYLCV)	4286	1071	
	Healthy (Po-H)	608	152	Healthy (TH)	1273	318	

Figure 4 visually represents some samples with class labels from the PlantVillage benchmark.

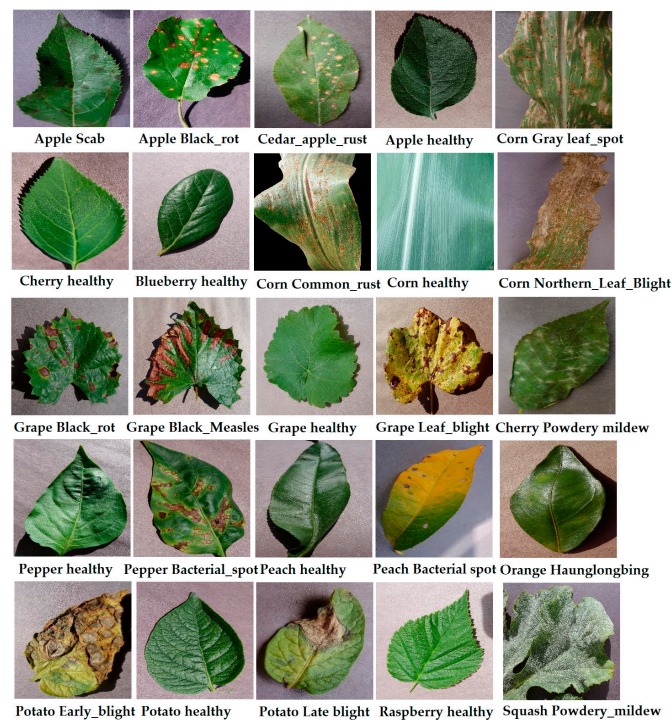


Figure 4. Cont.

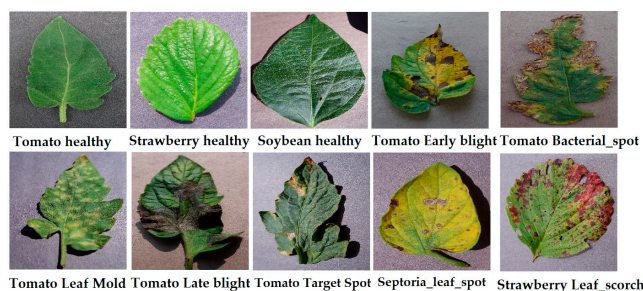


Figure 4. Some image samples obtained from the PlantVillage benchmark dataset.

3.4. Plant Disease Diagnosis Framework

3.4.1. The Principle of Baseline Model

The original baseline architecture (i.e., VGG-16) is shown in Figure 5. It consists of thirteen convolutional, five max-pooling, and three dense layers, which sum up to twenty-one layers. Among these 21 layers, only 16 are weight layers or learnable parameters layers (i.e., 13 convolutional layers and 3 dense layers). The VGG-16 model requires a 224×224 input image with three color channels. The VGG-16 architecture applies a 3×3 filter in the convolutional layers with stride and padding. Additionally, in the max-pooling layers, the stride is 2 and the filter size is 2×2 . The number of features on the map gradually increases (i.e., 64, 128, 256, 512, etc.), and the dimension of the feature maps reduces (i.e., 224×224 , 112×112 , 56×56 , 28×28 , 14×14 , 7×7 , etc.) in descending layers.

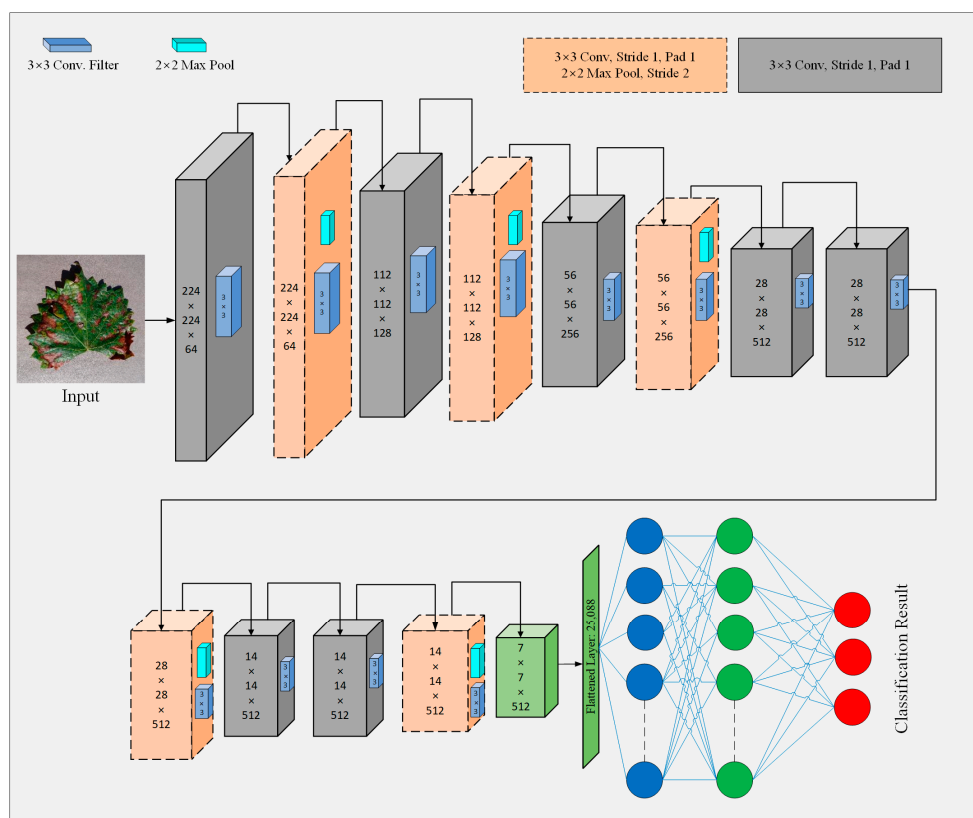


Figure 5. The VGG16 baseline model.

3.4.2. Model Improvements

The suggested CNN model is presented in Figure 6. The designed CNN architecture automatically extracts the semantic features, as VGG-16 does. However, VGG-16 ignores the multilevel and multiscale features. The proposed CNN architecture utilized dense connections and pyramid pooling for feature extraction. The first block of the designed

CNN framework comprises five convolutional layers adjacent to max-pooling layers. The purpose of this convolution and max-pooling block is to extract semantic features and perform dimensional reduction. Just like VGG-16, the initially proposed model applies convolution and max pooling to extract semantic features and reduce the dimensionality of feature maps. After the convolution and max-pooling blocks, a dense connection block is introduced to extract multilevel features. This dense connection block is inspired by Dense-Net, which utilizes the features of all previous layers in its descending layers. After multilevel feature extraction, a spatial pyramid pooling (SPP) block is introduced to extract multiscale features. The SPP block is inspired by a spatial pyramid pooling network. This SPP block helps identify plant infection at an early stage. After performing three different types of feature extraction, this study concatenates all these features for prediction.

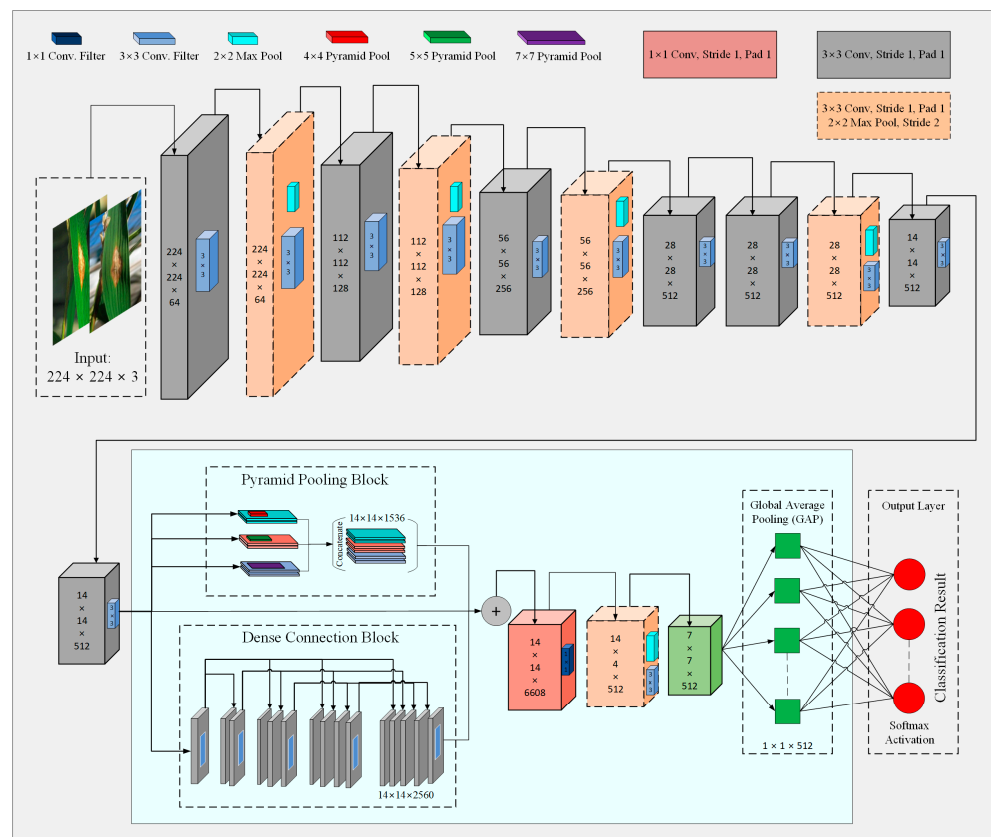


Figure 6. Proposed CNN backbone architecture.

The major components of the proposed model are as follows.

1. Multilevel features extraction

Exploring the limited capability of VGG-16 on feature extraction, we use the dense connection block (DCB) shown in Figure 6. The DCB is inspired by DenseNet121 [37], which enhances the baseline architecture to boost the feature extraction capability and maximize information flow within network layers.

The CNN model forward propagating the connection among the l^{-1} layer and the l th layer is described by Equation (1):

$$x^l = f(x^{l-1} * w^l + b^l) \tag{1}$$

where $f(\cdot)$ represents the FTS [41] as an activation function, x^{l-1} is the input of l th layer, w^l is the weight of convolution kernel, $*$ depicts the convolution, and b^l is a bias parameter.

The feature maps in the dense connection block of layer l^{-1} are concatenated and used as the input of the next l th layer described in Equation (2):

$$x^l = f([x^0, x^1, x^2, \dots, x^{l-1}] * w^l + b^l) \quad (2)$$

In DCB, each convolution layer outputs the concatenated feature maps that depict 'k', and the l th layer outputs the $[k_0 + k_1 + \dots + k_{l-1}]$ feature maps.

2. Multiscale features extraction

To boost the stability of the classifier for plant leaf lesion scale variation, this research introduces the pyramid pooling block (PPB) in the CNN architecture that utilizes local and global multiscale features to enhance the accuracy for small-scale lesion classification. The PPB comprises three max-pooling layers that pool feature maps at different scales using three distinct sliding windows. The sizes of the sliding windows are $\lceil Xdim/2 \rceil \times \lceil Ydim/2 \rceil$, $\lceil Xdim/3 \rceil \times \lceil Ydim/3 \rceil$, and $\lceil Xdim/4 \rceil \times \lceil Ydim/4 \rceil$, respectively. The PPB outputs the multiscale feature maps for the next convolutional layer.

3. Activation function

In CNN training, the activation function plays an essential role because it transforms the input signal into the output signal. The activation function introduces a nonlinearity factor for better classification. So, it is important to choose an efficient activation function that handles the nonlinearity of the training data with less complexity. Several traditional activation functions have been used for classification problems, such as Sigmoid, ReLu [46], Tanh, etc. Using the Sigmoid or Tanh activation functions, when we back-propagate through deeper layers, the propagation value reduces, which results in gradient disappearance. Moreover, the Sigmoid function is based on a complex power operation that slows down the training mechanism. Although the ReLu activation function has a strong convergence rate, it results in dead neurons and cannot squeeze the data point; thus, the scale of the data point will continue to increase as the number of layers grows. By considering these constraints on traditional activation functions, researchers proposed more stable and robust activation functions such as Swish [50] and Flatten-T Swish (FTS) [41]. In this research, we introduced the FTS activation function instead of ReLu in our proposed multilevel and multiscale CNN architecture. The FTS activation function is expressed in Equation (3), where Thr represents the threshold (i.e., -0.20):

$$FTS(x) = \begin{cases} \frac{x}{1 + e^{-x}} + Thr, & \text{if } x \geq 0 \\ Thr, & \text{if } x < 0 \end{cases} \quad (3)$$

The FTS activation function provides a high convergence rate during training and mitigates the gradient vanishing problem during back propagation.

4. Global Average Pooling (GAP)

After extracting rich semantic features, multiple fully connected or dense connection layers are used in the baseline architecture to stretch the long feature vector before feeding it to the SoftMax classifier. Fully connected layers use a large number of training parameters that slow down the training procedure, which results in overfitting. To mitigate this limitation of the baseline architecture, a GAP layer [23] is introduced to replace the fully connected layers. The GAP layer extracts a feature point for each feature map and constructs a feature vector to feed the SoftMax classifier. The GAP layer sums up the spatial information of feature maps and generates a more robust feature vector compared to a fully connected layer.

3.4.3. Model Fine Tuning and Loss Function

Using PlantVillage training samples and their corresponding labels, the proposed PDD-Net was fine tuned to minimize the loss function presented in Equation (4):

$$L(Y_{\text{act}}, f(x)) = -\frac{1}{N} \sum_{i=0}^N \sum_{j=0}^C \beta (1 - f_j)^\alpha y_{ij} \log(f_j; \theta) \quad (4)$$

Lin et al. [51] and Zhong et al. [25] first conceived of this loss function, they used this focal loss function to solve the category imbalance (i.e., between background and foreground) problem for the classification of plant diseases. Y_{act} is the actual label and $f(x)$ is the predicted value, β mitigates the category imbalance, and $\alpha > 0$ minimizes the loss of easily classifiable samples and increases the loss of difficult samples.

3.4.4. Model Performance Metrics

The performance of the proposed PDD-Net method and other CNN classifiers is calculated using different metrics such as recall, precision, accuracy, and F1 score. As we report the performance metrics for each class label, we explain the calculations obtained from the combined confusion matrices, as follows. Let $C_{i,j}$ be an element in the combined confusion matrices at the i th row and j th column. For a given class label at the i th row, the number of true positive samples ($TP = C_{i,i}$), the number of false positive samples ($FP = \sum_{j \neq i} C_{j,i}$), and the number of false negative samples ($FN = \sum_{j \neq i} C_{i,j}$) are given. Equations (5)–(7) show the performance metrics for each class as

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{and F1 score} = 2 \times \frac{(\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})} \quad (7)$$

For a given method, we also report the average performance results of all classes. The accuracy performance measure is calculated as

$$\text{Accuracy} = \frac{\sum_i C_{i,i}}{\sum_i \sum_j C_{i,j}} \quad (8)$$

3.5. Model Training and Testing

The proposed deep-learning-based PDD-Net was trained on various leaf images of different plants for plant disease classification. The model training and testing were performed using an Intel Xenon processor, 64 GB of RAM, and an NVIDIA-TITAN-RTX-GPU. During model training, the input image dimension was 224×224 with a batch size of 32. The SGD was used as an optimizer with a weight decay of 0.0005, learning rate of 0.001, and momentum of 0.9 to train the proposed PDD-Net. The SGD was used instead of the Adam optimizer because of its high performance [52].

We utilized fivefold cross-validation, where the test set consisted of 20% of the total samples while the leftovers (i.e., 80% data samples) were used for training. In each run of the fivefold cross-validation, we performed predictions to test the samples. As the confusion matrix compares the true labels against predictions, we combined the confusion matrices in all five runs of fivefold cross-validation and then calculated the performance results.

4. Results

4.1. PlantVillage

Here, we report the prediction performance of PDD-Net on the PlantVillage dataset. Figure 7 corresponds to the combined confusion matrices for all 38 classes on unseen 12,784 test images. Several other CNN-based classifiers, including DenseNet-201 [21,22], DenseNet-121 [22,25,29,33], ResNet-50 [21,30,33], and VGG-16 [26,32,33] were also trained on the same samples. The confusion matrix of each model was generated and combined, and their corresponding true positive rates were compared with the proposed model; the comparison is shown in Figure 8 for all classes of the PlantVillage dataset. The model evaluation matrices (i.e., precision, recall, F1 score, and accuracy) are directly proportional to true positive values. The true positive values of the proposed hybrid model were superior to the baseline and other architectures used for plant disease classification, utilizing multi-level and multiscale high-level features.

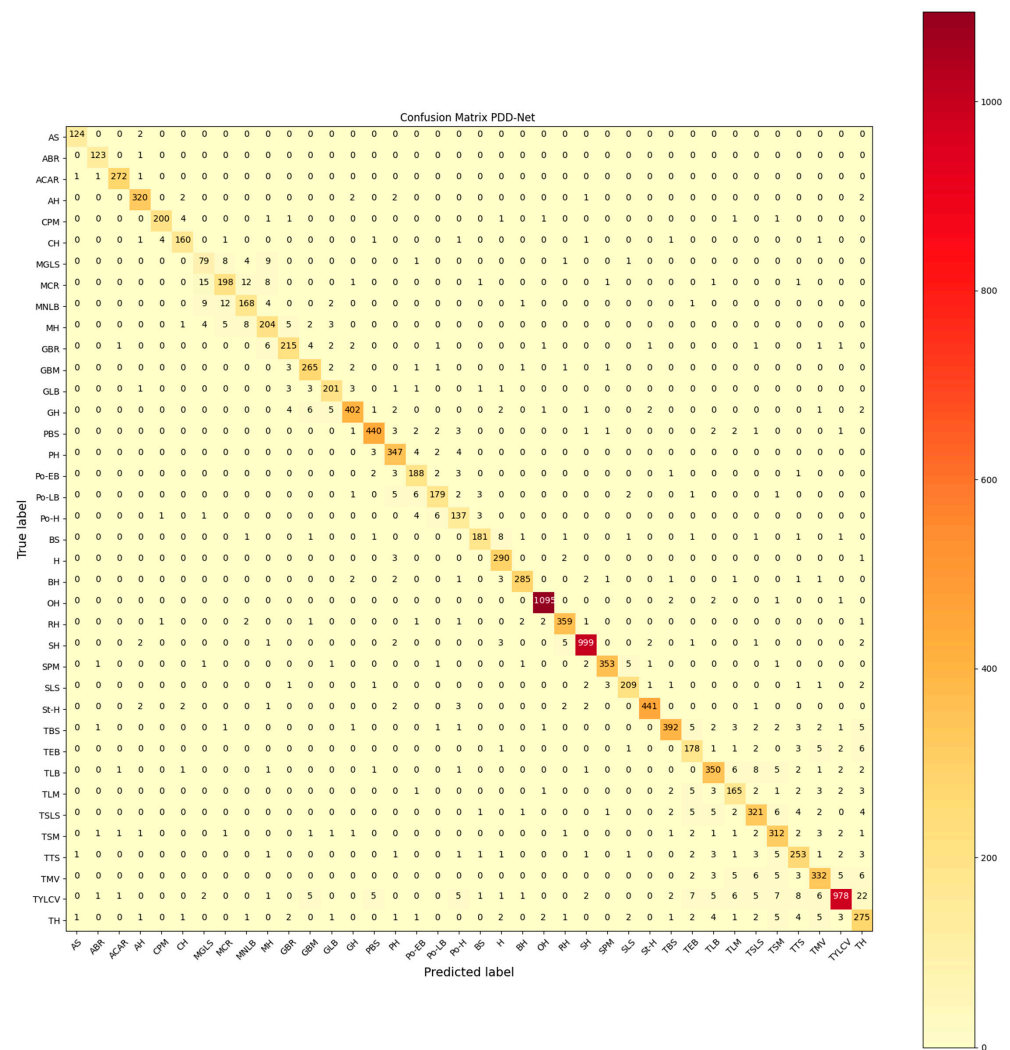


Figure 7. Combined confusion matrices of PDD-Net using PlantVillage test sets in the 5 runs of fivefold cross-validation.

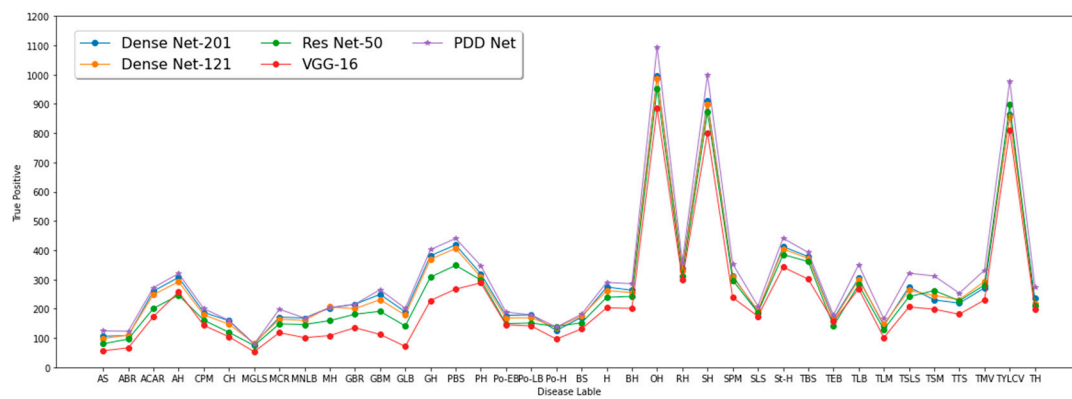


Figure 8. Comparison of true positive values for each class.

We calculate the precision, recall, and F1 score of the proposed PDD-Net, as shown in Table 3, for all 38 classes of the PlantVillage benchmark dataset.

Table 3. Classification performance of PDD-Net for each class pertaining to the PlantVillage benchmark dataset.

Class	Precision %	Recall %	F1 Score %
AS	97.64	98.41	98.02
ABR	96.09	99.19	97.62
ACAR	98.55	98.91	98.73
AH	96.39	97.26	96.82
CPM	97.09	95.24	96.15
CH	93.57	93.57	93.57
MGLS	71.17	76.70	73.83
MCR	87.61	83.19	85.34
MNLB	85.71	85.28	85.50
MH	86.08	87.93	86.99
GBR	91.88	91.10	91.49
GBM	92.01	95.67	93.81
GLB	92.20	93.49	92.84
GH	96.17	93.71	94.92
PBS	96.70	95.86	96.28
PH	92.78	96.39	94.55
Po-EB	89.52	94.00	91.71
Po-LB	91.79	89.50	90.63
Po-H	84.05	90.13	86.98
BS	94.27	90.95	92.58
H	92.65	97.97	95.24
BH	97.27	95.00	96.12
OH	99.18	99.46	99.32
RH	96.25	96.77	96.51
SH	98.33	98.13	98.23
SPM	97.78	96.19	96.98
SLS	94.14	94.14	94.14
St-H	98.44	96.71	97.57
TBS	96.55	92.67	94.57
TEB	83.96	89.00	86.41
TLB	91.62	91.62	91.62
TLM	84.62	86.84	85.71
TSLs	89.66	90.68	90.17
TSM	88.64	93.13	90.83
TTS	87.54	90.04	88.77
TMV	90.96	90.46	90.71
TYLCV	97.70	91.32	94.40
TH	81.60	86.48	83.97

The overall performance was evaluated by taking the average of all classes. We compared the overall performance of PDD-Net with several CNN-based classifiers used for plant disease classification, including DenseNet-201 [21,22], DenseNet-121 [22,25,29,33], ResNet-50 [21,30,33], and VGG-16 [26,32,33].

Table 4 depicts the overall average performance (excluding the accuracy that was calculated without taking the average) of different CNN architectures. DenseNet-201 achieved an average precision of 82.82%, average recall of 85.13%, average F1 score of 83.82%, and accuracy of 85.60%. DenseNet-121 achieved an average precision of 81.30%, average recall of 83.47%, average F1 score of 82.15%, and accuracy of 84.05%. ResNet-50 achieved an average precision of 75.00%, average recall of 76.13%, average F1 score of 75.19%, and accuracy of 78.38%. VGG-16 achieved an average precision of 60.87%, average recall of 63.36%, average F1 score of 61.67%, and accuracy of 67.12%. Additionally, our proposed PDD-Net achieved the highest average precision of 92.06%, highest average recall of 92.71%, highest average F1 score of 92.36%, and highest accuracy of 93.79%. Overall, these results show the superiority of PDD-Net compared to other models. ResNet-50 and VGG-16 exhibited a poorer performance compared to other frameworks.

Table 4. Average performance comparison of CNN-based frameworks on PlantVillage dataset. The model with the best performance results is shown in bold. Accuracy is calculated using Equation (8).

Framework	Precision %	Recall %	F1 Score %	Accuracy %
DenseNet-201	82.82	85.13	83.82	85.60
DenseNet-121	81.30	83.47	82.15	84.05
ResNet-50	75.00	76.13	75.19	78.38
VGG-16	60.87	63.36	61.67	67.12
PDD-Net	92.06	92.71	92.36	93.79

4.2. Cassava Leaf Disease (CLD)

This experiment was performed by using a CLD benchmark dataset. The combined confusion matrix (depicted in Figure 9) was analyzed to assess the performance of the proposed PDD-Net.

Based on the combined confusion matrices, the estimated precision, recall, and F1 score of PDD-Net for each class of CLD are shown in Table 5. For HCL, the model achieved a precision of 90%, a recall of 87.31%, and an F1 score of 88.64%. In classifying CBB, PDD-Net attained a precision of 82.28%, a recall of 77.74%, and an F1 score of 79.94%. The PDD-Net model exhibited similar performance results when detecting CBSD with a precision of 84.51%, a recall of 84.51%, and an F1 score of 84.51%. For CMD, the most common disease in the CLD dataset, PDD-Net performed exceptionally well, achieving a precision of 86.46%, a recall of 94.60%, and an F1 score of 90.35%. Lastly, the model demonstrated a good performance when detecting Cassava Green Mottle (CGM) with a precision of 88.79%, a recall of 84.66%, and an F1 score of 86.68%. These results highlight the efficiency and effectiveness of the PDD-Net architecture in addressing the complex task of plant disease diagnosis.

Table 5. Classification performance of PDD-Net for each class of the CLD dataset.

Category	Precision %	Recall %	F1 Score %
HCL	90.00	87.31	88.64
CBB	82.28	77.74	79.94
CBSD	84.51	84.51	84.51
CMD	86.46	94.60	90.35
CGM	88.79	84.66	86.68

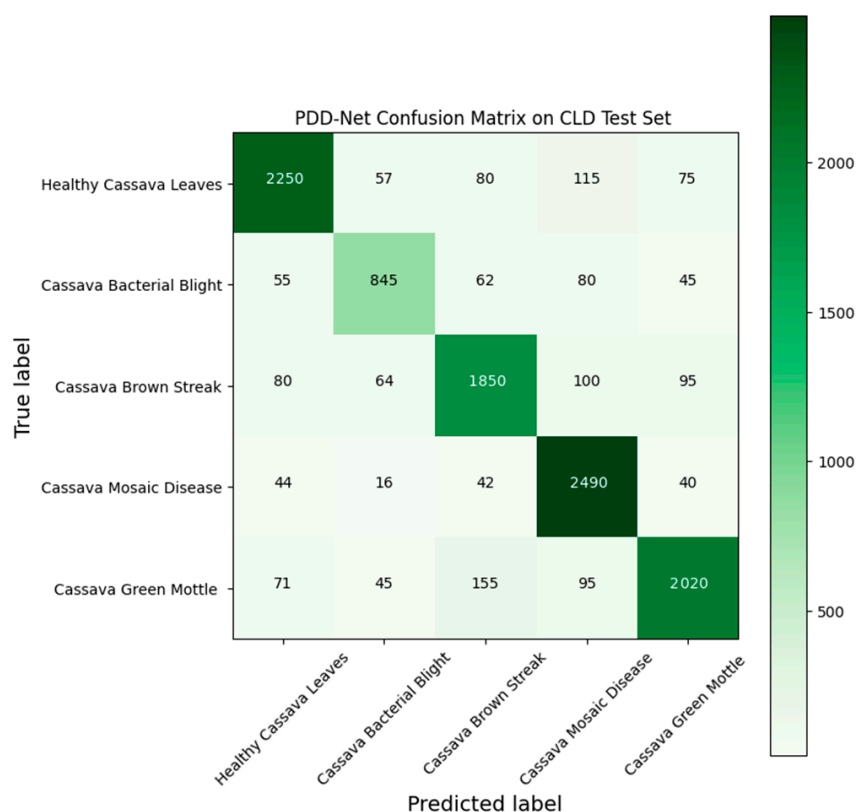


Figure 9. Combined confusion matrices of PDD-Net using “CLD” test set in the 5 runs of fivefold cross-validation.

A detailed comparison of the averaged results with other state-of-the-art models such as DenseNet-201, DenseNet-121, ResNet-50, and VGG-16 is presented in Table 6 to demonstrate the effectiveness of PDD-Net. The analysis revealed that PDD-Net achieved the highest average precision (86.41%) among the compared models, indicating its superior ability to accurately distinguish between different cassava leaf diseases and healthy samples. In contrast, DenseNet-201, DenseNet-121, ResNet-50, and VGG-16 exhibited lower average precision values of 81.77%, 80.76%, 79.92%, and 78.93%, respectively. This highlights the importance of PDD-Net’s design choices in achieving high-performance results.

Table 6. Average performance comparison of CNN-based frameworks on the CLD dataset. The method with best performance results is shown in bold. Accuracy is calculated using Equation (8).

Framework	Precision %	Recall %	F1 Score %	Accuracy %
DenseNet-201	81.77	81.76	81.57	82.50
DenseNet-121	80.76	80.35	80.33	81.27
ResNet-50	79.92	79.53	79.55	80.88
VGG-16	78.93	78.54	78.54	79.89
PDD-Net	86.41	85.77	86.02	86.98

Furthermore, PDD-Net achieved the highest recall (85.77%) among the models, emphasizing its effectiveness at identifying true positive instances within the dataset. The competing models, DenseNet-201, DenseNet-121, ResNet-50, and VGG-16, demonstrated lower average recall values of 81.76%, 80.35%, 79.53%, and 78.54%, respectively. The higher recall value for PDD-Net signifies its potential to minimize missed diseased leaf instances, which is crucial for effective disease management.

In terms of the F1 score, a metric that considers both precision and recall, PDD-Net outperformed the other models with an average F1 score of 86.02%. In comparison, DenseNet-201, DenseNet-121, ResNet-50, and VGG-16 achieved lower average F1 scores of

81.57%, 80.33%, 79.55%, and 78.54%, respectively, emphasizing the superior performance of PDD-Net.

Lastly, PDD-Net outperformed the other models in terms of accuracy, with a value of 86.98%. This attests to the model's high capability to correctly classify instances of cassava leaf disease. The DenseNet-201, DenseNet-121, ResNet-50, and VGG-16 models reported lower accuracy values of 82.50%, 81.27%, 80.88%, and 79.89%, respectively. The superior accuracy of PDD-Net demonstrates its potential for practical applications in agriculture.

The proposed PDD-Net exhibited superior performance on two different benchmarks compared to VGG-16, ResNet-50, DenseNet-121, and DenseNet-201. The benefits of PDD-Net's design selections, including multilevel and multiscale features, FTS activation function, and focal loss function to address class imbalance, contribute to its enhanced performance. These reported results make PDD-Net a promising solution for practical applications in agriculture.

5. Discussion

The proposed framework, PDD-Net, is designed to efficiently diagnose plant diseases using a CNN based on the VGG-16 architecture. This framework aims to mitigate the limitations of existing CNN-based architectures by investigating the influence of multilevel and multiscale features in plant disease diagnosis. For fast convergence and efficient model training, a state-of-the-art activation function known as FTS is used to avoid the dead neuron problem in backpropagation. To enhance data sample diversity and mitigate the influence of overfitting, data augmentation techniques are utilized. Data augmentation and the focal loss function employed in PDD-Net help with class imbalance problems in benchmarks.

The PDD-Net method was evaluated against other baseline methods using two plant disease classification benchmark datasets: PlantVillage and cassava leaf disease (CLD). The results in Tables 4 and 6 reveal that PDD-Net outperformed VGG-16, ResNet-50, DenseNet-121, and DenseNet-201 in terms of accuracy, precision, recall, and F1 score. The baseline architecture utilized 138.35 million training parameters, while the proposed PDD-Net required only 16.67 million because PDD-Net used a GAP layer instead of fully connected layers.

Various software tools were used to implement and evaluate the proposed model, such as TensorFlow and Keras, which were used to build the CNN architecture, and the Python programming language for data processing and analysis [53]. These tools provide a robust and flexible environment for developing and testing deep learning models. The optimizer in DL plays a key role in updating model weights. The SGD optimizer with our framework achieved better results than the ADAM optimizer. When training our framework, we used a learning rate of 0.001 and momentum of 0.9.

6. Conclusions and Future Work

We present a multilevel and multiscale CNN architecture, PDD-Net, to improve the prediction performance in leaf disease diagnosis. The PDD-Net method allowed for the detection of fine-grained visual patterns in plant leaves at different levels of abstraction, which resulted in high-performance results when identifying various diseases while handling class variations. While the multiscale approach enhances the network's capability to identify small-scale variations in images, it allowed diseases to be diagnosed at an early stage. Along with multilevel and multiscale features, using GAP in CNNs for plant disease classification can lead to accurate and robust models that are less likely to overfit the training data with fewer parameters. Furthermore, the use of transfer learning, where pretrained CNN models were fine-tuned on the PlantVillage and CLD datasets, resulted in faster convergence and better accuracy than training from scratch. Additionally, our approach leveraged the knowledge obtained from large-scale datasets and optimized the CNN models for plant disease diagnosis. When testing the performance on the PlantVillage dataset, PDD-Net achieved the highest average precision of 92.06%, the highest average recall of 92.71%, the highest average F1 score of 92.36%, and the highest accuracy of 93.79%.

Similarly, when testing the performance on the cassava leaf disease dataset, PDD-Net achieved the highest average precision of 86.41%, the highest average recall of 85.77%, the highest average F1 score of 86.02%, and the highest accuracy of 86.98%. These results show that DL can help farmers, improve crop yield, and contribute to global food security.

Some challenges must be addressed in future work developing CNN models for plant disease diagnosis, such as (1) utilizing boosting techniques as mentioned in [54] to improve the prediction performance, and (2) adapting the existing framework to perform predictions and feature extraction, followed by assessing the performance when these features are coupled with machine learning methods for problems in biology and medicine, as described in [53].

Author Contributions: T.T. and H.A. conceived and designed the study. H.A. performed the analysis. H.A. wrote the manuscript. T.T. and H.A. revised and edited the manuscript. T.T. supervised the study. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used are cited within the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Taghikhah, F.; Voinov, A.; Shukla, N.; Filatova, T.; Anufriev, M. Integrated modeling of extended agro-food supply chains: A systems approach. *Eur. J. Oper. Res.* **2021**, *288*, 852–868. [\[CrossRef\]](#)
2. Imami, D.; Valentinov, V.; Skreli, E. Food Safety and Value Chain Coordination in the Context of a Transition Economy: The Role of Agricultural Cooperatives. *Int. J. Commons* **2021**, *15*, 21–34. [\[CrossRef\]](#)
3. Shang, Y.; Hasan, M.K.; Ahammed, G.J.; Li, M.; Yin, H.; Zhou, J. Applications of Nanotechnology in Plant Growth and Crop Protection: A Review. *Molecules* **2019**, *24*, 2558. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Bass, D.; Stentiford, G.D.; Wang, H.-C.; Koskella, B.; Tyler, C.R. The Pathobiome in Animal and Plant Diseases. *Trends Ecol. Evol.* **2019**, *34*, 996–1008. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant Disease Detection and Classification by Deep Learning. *Plants* **2019**, *8*, 468. [\[CrossRef\]](#)
6. Nagaraju, M.; Chawla, P. Systematic review of deep learning techniques in plant disease detection. *Int. J. Syst. Assur. Eng. Manag.* **2020**, *11*, 547–560. [\[CrossRef\]](#)
7. Shruthi, U.; Nagaveni, V.; Raghavendra, B.K. A Review on Machine Learning Classification Techniques for Plant Disease Detection. In Proceedings of the 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 15–16 March 2019; pp. 281–284.
8. Pantazi, X.; Moshou, D.; Tamouridou, A. Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers. *Comput. Electron. Agric.* **2019**, *156*, 96–104. [\[CrossRef\]](#)
9. Hlaing, C.S.; Zaw, S.M.M. Tomato Plant Diseases Classification Using Statistical Texture Feature and Color Feature. In Proceedings of the 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), Singapore, 6–8 June 2018; pp. 439–444.
10. Muhathir, M.; Hidayah, W.; Ifantiska, D. Utilization of Support Vector Machine and Speeded up Robust Features Extraction in Classifying Fruit Imagery. *Comput. Eng. Appl. J.* **2020**, *9*, 183–193. [\[CrossRef\]](#)
11. Iniyani, S.; Jebakumar, R.; Mangalraj, P.; Mohit, M.; Nanda, A. Plant Disease Identification and Detection Using Support Vector Machines and Artificial Neural Networks. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*; Springer: Singapore, 2020; pp. 15–27.
12. Mohameth, F.; Bingcai, C.; Sada, K.A. Plant Disease Detection with Deep Learning and Feature Extraction Using Plant Village. *J. Comput. Commun.* **2020**, *8*, 10–22. [\[CrossRef\]](#)
13. Hanbay, K. Hyperspectral image classification using convolutional neural network and two-dimensional complex Gabor transform. *J. Fac. Eng. Archit. Gazi Univ.* **2020**, *35*, 443–456.
14. Kusumo, B.S.; Heryana, A.; Mahendra, O.; Pardede, H.F. Machine Learning-Based for Automatic Detection of Corn-Plant Diseases Using Image Processing. In Proceedings of the 2018 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), Tangerang, Indonesia, 1–2 November 2018; pp. 93–97.
15. Golhani, K.; Balasundram, S.K.; Vadamalai, G.; Pradhan, B. A review of neural networks in plant disease detection using hyperspectral data. *Inf. Process. Agric.* **2018**, *5*, 354–371. [\[CrossRef\]](#)

16. Tianyu, Z.; Zhenjiang, M.; Jianhu, Z. Combining cnn with Hand-Crafted Features for Image Classification. In Proceedings of the 2018 14th IEEE International Conference on Signal Processing (ICSP), Beijing, China, 12–16 August 2018; pp. 554–557.
17. Fooladgar, F.; Kasaei, S. A survey on indoor RGB-D semantic segmentation: From hand-crafted features to deep convolutional neural networks. *Multimedia Tools Appl.* **2020**, *79*, 4499–4524. [[CrossRef](#)]
18. Peng, X.; Yu, J.; Yao, B.; Liu, L.; Peng, Y. A Review of FPGA-Based Custom Computing Architecture for Convolutional Neural Network Inference. *Chin. J. Electron.* **2021**, *30*, 1–17.
19. Lu, J.; Tan, L.; Jiang, H. Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification. *Agriculture* **2021**, *11*, 707. [[CrossRef](#)]
20. Zhang, Y.-D.; Dong, Z.; Chen, X.; Jia, W.; Du, S.; Muhammad, K.; Wang, S.-H. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools Appl.* **2019**, *78*, 3613–3632. [[CrossRef](#)]
21. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanekaran, Y. Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* **2020**, *173*, 105393. [[CrossRef](#)]
22. Tiwari, V.; Joshi, R.C.; Dutta, M.K. Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images. *Ecol. Inform.* **2021**, *63*, 101289. [[CrossRef](#)]
23. Zhang, S.; Zhang, S.; Zhang, C.; Wang, X.; Shi, Y. Cucumber leaf disease identification with global pooling dilated convolutional neural network. *Comput. Electron. Agric.* **2019**, *162*, 422–430. [[CrossRef](#)]
24. Zhang, J.; Rao, Y.; Man, C.; Jiang, Z.; Li, S. Identification of cucumber leaf diseases using deep learning and small sample size for agricultural Internet of Things. *Int. J. Distrib. Sens. Netw.* **2021**, *17*, 15501477211007407. [[CrossRef](#)]
25. Zhong, Y.; Zhao, M. Research on deep learning in apple leaf disease recognition. *Comput. Electron. Agric.* **2020**, *168*, 105146. [[CrossRef](#)]
26. Wang, J.; Yu, L.; Yang, J.; Dong, H. DBA_SSD: A Novel End-to-End Object Detection Algorithm Applied to Plant Disease Detection. *Information* **2021**, *12*, 474. [[CrossRef](#)]
27. Chakraborty, K.K.; Mukherjee, R.; Chakraborty, C.; Bora, K. Automated recognition of optical image based potato leaf blight diseases using deep learning. *Physiol. Mol. Plant Pathol.* **2022**, *117*, 101781. [[CrossRef](#)]
28. Wagle, S.A.; Harikrishnan, R.; Ali, S.H.M.; Faseehuddin, M. Classification of Plant Leaves Using New Compact Convolutional Neural Network Models. *Plants* **2021**, *11*, 24. [[CrossRef](#)]
29. Abbas, A.; Jain, S.; Gour, M.; Vankudothu, S. Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Comput. Electron. Agric.* **2021**, *187*, 106279. [[CrossRef](#)]
30. Tariq, U.; Hussain, N.; Nam, Y.; Kadry, S. An Integrated Deep Learning Framework for Fruits Diseases Classification. *Comput. Mater. Contin.* **2022**, *71*, 1387–1402.
31. Thompson, R.N.; Brooks-Pollock, E. Detection, forecasting and control of infectious disease epidemics: Modelling outbreaks in humans, animals and plants. *Philos. Trans. R Soc. B Biol. Sci.* **2019**, *374*, 20190038. [[CrossRef](#)]
32. Alatawi, A.A.; Alomani, S.M.; Alhawiti, N.I.; Ayaz, M. Plant Disease Detection using AI based VGG-16 Model. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*. [[CrossRef](#)]
33. Eunice, J.; Popescu, D.E.; Chowdary, M.K.; Hemanth, J. Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications. *Agronomy* **2022**, *12*, 2395.
34. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
35. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
36. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
37. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
38. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
40. Msonda, P.; Uymaz, S.A.; Karaağaç, S.S. Spatial Pyramid Pooling in Deep Convolutional Networks for Automatic Tuberculosis Diagnosis. *Trait. Du Signal* **2020**, *37*, 1075–1084. [[CrossRef](#)]
41. Chieng, H.H.; Wahid, N.; Ong, P.; Perla, S.R.K. Flatten-T Swish: A thresholded ReLU-Swish-like activation function for deep learning. *arXiv* **2018**, arXiv:1812.06247. [[CrossRef](#)]
42. Cap, Q.H.; Uga, H.; Kagiwada, S.; Iyatomi, H. LeafGAN: An Effective Data Augmentation Method for Practical Plant Disease Diagnosis. *IEEE Trans. Autom. Sci. Eng.* **2020**, *19*, 1258–1267. [[CrossRef](#)]
43. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
44. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

45. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single Shot Multibox Detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer International Publishing: Amsterdam, The Netherlands, 2016; Part I 14, pp. 21–37.
46. Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
47. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
48. Attallah, O. Tomato Leaf Disease Classification via Compact Convolutional Neural Networks with Transfer Learning and Feature Selection. *Horticulturae* **2023**, *9*, 149. [[CrossRef](#)]
49. Koo, K.-M.; Cha, E.-Y. Image recognition performance enhancements using image normalization. *Hum. Cent. Comput. Inf. Sci.* **2017**, *7*, 33. [[CrossRef](#)]
50. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
51. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2999–3007.
52. Hardt, M.; Recht, B.; Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In Proceedings of the International Conference on Machine Learning, PMLR. New York City, NY, USA, 19–24 June 2016; pp. 1225–1234.
53. Turki, T.; Taguchi, Y.H. Discriminating the Single-cell Gene Regulatory Networks of Human Pancreatic Islets: A Novel Deep Learning Application. *Comput. Biol. Med.* **2021**, *132*, 104257. [[CrossRef](#)] [[PubMed](#)]
54. Turki, T.; Wei, Z. Improved Deep Convolutional Neural Networks via Boosting for Predicting the Quality of In Vitro Bovine Embryos. *Electronics* **2022**, *11*, 1363. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.