



Cost-optimized hybrid convolutional neural networks for detection of plant leaf diseases

Adem Tuncer¹

Received: 24 June 2020 / Accepted: 28 April 2021 / Published online: 8 May 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

Convolutional neural network (CNN) models have been used extensively in many image recognition tasks for their state-of-the-art performance in recent years. Researchers inspired by this success frequently prefer CNNs in the agricultural field, especially for disease detection and classification. Many CNN models have been proposed for plant leaf diseases and impressive performance results have been obtained. On the other hand, standard CNNs usually need millions of parameters in the network for computation, but it is difficult to implement them on embedded and mobile devices with limited resources. Therefore, it is important to obtain lighter models by decreasing the number of parameters in addition to the high performance of the models. In this paper, a new hybrid CNN approach based on Inception architecture and depthwise separable convolutions is proposed to reduce the number of parameters and computational cost for plant leaf disease detection and classification. Although the number of parameters is significantly reduced, the results show that the proposed approach has high accuracy performance. The proposed hybrid model has been trained and tested with k -fold cross-validation using a dataset of 50,136 images containing 30 classes from 14 different leaves, including healthy and diseased ones. The new model has achieved the best accuracy of 99.27% and an average accuracy of 99%, and provides about a 75% reduction in the number of parameters compared to the standard CNN.

Keywords Convolutional neural network · Computation cost · Depthwise separable convolution · Inception · Plant leaf disease detection

1 Introduction

Deep learning implies artificial neural networks, which, unlike standard neural network architectures, include numerous processing layers. Deep learning approaches have seen significant improvements in recent years. CNNs are the most popular of deep neural networks because of their great success in the classification of large-scale image datasets (Jin et al. 2019). In recent years, they have achieved excellent performance and state-of-the-art results in many fields, such as image classification and clustering and pattern recognition applications. They have drastically changed traditional image processing methods, and in this respect, have become increasingly popular in many image processing applications. It is possible to extract the appropriate features from training

datasets automatically with CNNs instead of relying on manual feature extraction (Ma et al. 2018). Various studies have made significant improvements in classification accuracy by trying different filter sizes or different network depths in CNN models. In particular, AlexNet (Krizhevsky et al. 2012), VGGNet (Simonyan and Zisserman 2014), Inception (Szegedy et al. 2015), and ResNet (He et al. 2016) are some of the popular CNN architectures.

Recent studies on leaf diseases show that plant diseases affect the growth and crop yield of the plants. This problem causes social, ecological, and economic impacts on agriculture. Detection and classification of plant leaf diseases are important issues in agricultural research. The detection is traditionally carried out by human experts. Human experts identify diseases visually, but they face some difficulties that may harm their efforts. This approach is common in practice, but it is error-prone and expensive in terms of time and labor usage (Barbedo 2016). There are studies in the literature to automate detection and classification. However, the solutions proposed so far suffer from limitations due

✉ Adem Tuncer
adem.tuncer@yalova.edu.tr

¹ Department of Computer Engineering, Yalova University, Yalova, Turkey

to the methods used and datasets. As in many fields, deep learning-based approaches also in the agriculture field are extensively explored in recent years. The image classification solutions based on CNN have been successfully applied to plant leaf disease detection, crop, fruit classification, and weed identification in the agricultural field (Kamilaris and Prenafeta-Boldú 2018).

Real-time object detection applications on embedded or mobile devices are becoming increasingly popular using deep learning algorithms. Due to the fact that CNN-based algorithms can be employed on these devices in recent years, it has become easier and more practical to detect plant leaf diseases in real-time in agricultural areas. However, the resources on embedded devices are limited such as computing capability and storage capacity. The number of parameters used in a network and the computation cost vary depending on the size and complexity of the problem. CNNs usually need millions of parameters in the network for computation. Therefore, they have large computational complexity and are difficult to deploy on embedded devices with limited resources (Zhang et al. 2019a). As the number of parameters in the network increases, it is very important to reduce computation cost for real-time applications.

This paper proposes a hybrid deep learning architecture with fewer parameters for the detection and classification of different plant leaves. The Inception architecture and depthwise separable convolutional neural network have been combined, and the hyper-parameters have been tuned to achieve higher accuracy. The proposed model provides an advantage over standard CNN by reducing the number of parameters and computational complexity. The model has been trained using the PlantVillage (Hughes and Salathé 2015) dataset of 50,136 images containing 30 classes from 14 different leaves, including healthy and diseased ones, and the performance results are presented herein. The main contributions of this study are summarized as follows:

- A new architecture has been designed that gives high-performance results, by also avoiding overfitting problem for leaf diseases rather than pre-trained architectures.
- A hybrid approach is presented by combining the parallel structure of the Inception architecture and the advantage of depthwise separable convolutions.
- The proposed architecture is lighter with fewer parameters and faster compared to standard CNNs.
- The proposed architecture, which significantly reduces the number of parameters, can be implemented for real-time object detection applications on embedded and mobile devices with limited resources.
- Performance results show that although the proposed approach uses fewer parameters, it has the same success rate as other studies in the literature that offer high accuracy.

The rest of the paper is organized as follows: related works carried out on the agriculture field using CNN models are given in Sect. 2. In this section, studies also with depthwise separable convolutions are presented. Section 3 presents the materials and methods to accomplish the task of the proposed architecture. Experimental results are shown and discussed in Sect. 4, and the conclusion is presented in Sect. 5.

2 Related works

The literature has some studies on deep learning models developed for plant leaf disease detection. Most of the studies employ pre-trained, general-purpose models and some of them propose a modified version of standard CNN models to classify diseased leaves.

The pre-trained, general-purpose image recognition CNN architectures VGG, AlexNet, AlexNetOWTBn, Overfeat, and GoogLeNet were tested for their ability to identify plant diseases and compared by Ferentinos (2018). Pawara et al. (2017) presented a comparative study of different image recognition techniques, such as AlexNet and GoogLeNet, on three different plant leaf dataset. For their part, Lee et al. (2017) introduced a hybrid global-local feature extraction model for leaf data based on pre-trained CNN models and a deconvolutional network. In the research conducted by Durmuş et al. (2017), they trained and tested the AlexNet and SqueezeNet pre-trained network architectures on a dataset of tomato images. In their study, Zhang et al. (2018) presented modified versions of the GoogLeNet and Cifar10 models for maize leaf disease recognition. The models have been carried out with different pooling combinations and dropout operations. Rangarajan et al. (2018) used pre-trained deep learning architectures namely VGG16 and AlexNet for classifying tomato crop diseases with the images from the PlantVillage dataset including 6 diseases and a healthy class. 13,262 tomato images were used in the study and classification accuracy was 97.29% for VGG16 and 97.49% for AlexNex. Mohanty et al. (2016) trained AlexNet and GoogLeNet to recognize 14 crop species and 26 diseases. Too et al. (2019) reported a comparative study of four different convolutional neural network models including VGG 16, Inception V4, ResNet with 50, 101, and 152 layers, and DenseNets with 121 layers. They used the models for the classification of plant diseases consist of diseased classes and 14 healthy classes taken from the PlantVillage dataset. In the paper by Rangarajan Aravind and Raja (2020), the disease classification system was proposed for ten different diseases of four varieties of crops. AlexNet, VGG16, VGG19, GoogLeNet, ResNet101, and DenseNet201 which are the pre-trained deep learning models were evaluated. From comparisons, it was stated that GoogleNet had the best with 97.3% accuracy. Chen et al. (2020) studied the

transfer learning approach for plant leaf disease identification using pre-trained VGGNet and Inception architectures. They modified the pre-trained VGGNet by replacing its last layers with an additional convolutional layer and indicated that their approach reached an average accuracy of 92% for the classification of rice plant images. Hu et al. (2019) used VGG16 deep learning model to identify the tea leaf’s diseases, and indicated that the model reached an accuracy of 90%. In sum, these papers employed various well-known, pre-trained models as opposed to proposing a new CNN architecture, and most of these studies have focused on high detection accuracy, but there are not enough lighter models with less computation cost and time for embedded devices.

On the other hand, Lu et al. (2017) carried out a model for detection of rice diseases based on CNN. They indicated that their model could classify ten common rice diseases through image recognition, and their model achieved an accuracy of 95.48%. They compared their model with traditional machine learning algorithms, such as the standard back propagation algorithm, support vector machine, and particle swarm optimization. Ma et al. (2018) proposed symptom-wise recognition of four cucumber diseases based on a CNN model. The accuracy results of their system on unbalanced and balanced datasets were 93.4% and 92.2%, respectively. Zhang et al. (2019b) proposed a vegetable disease recognition approach based on three-channel CNN (TCCNN). The research by Sardogan et al. (2018) proposed the detection and classification of tomato leaves based on CNN with the learning vector quantization (LVQ) algorithm. Their CNN model used only one filter, and the fully connected layer was implemented using the LVQ algorithm. Huynh et al. (2020) proposed a CNN model for leaves classification and the leaf pre-processing extract modified for the red color channel was based on vein shape data. They used Flavia and

the Swedish dataset and reported that their model was effective with the best accuracy greater than 98.22%. Kaya et al. (2019) demonstrated the effect of the four transfer learning models for deep learning based on plant classification. They presented five general schemas for experimental studies consisting of end-to-end CNN, fine-tuning, cross dataset fine-tuning, deep feature learning, and CNN-RNN classification. Geetharamani and Pandian (2019) proposed a CNN model with a nine-layer to identify plant leaf diseases including 39 classes with a 96.46% accuracy rate. Singh et al. (2019) proposed multilayer CNN for the classification of mango leaves using PlantVillage and real-time captured datasets. They indicated that the proposed model achieved an accuracy of 97.13%.

There are also some related studies based on depthwise separable convolutions. Kamal et al. (2019) presented two versions of depthwise separable convolution comprising two varieties of building blocks using a subset of publicly available PlantVillage dataset. They compared their models to VGG and MobileNet. A video smoke detection algorithm for a forest fire by Peng and Wang (2019) and a lightweight face recognition by Li et al. (2019) was proposed based on depthwise separable convolutions. A channel pruning algorithm for depthwise separable convolution and a new channel selection, implemented on MobileNet were proposed by Zhang et al. (2019a).

Table 1 summarizes the related studies especially based on plant leaf diseases. Although the references have different results with different experiments, the best results obtained are given in the table. The best and average accuracy values obtained in this study are comparable to the recent studies. On the other hand, the proposed model achieves these successful results with fewer parameters decreasing the computation cost.

Table 1 The summary of related studies

References	Method	Datasets	Accuracy
Mohanty et al. (2016)	GoogleNet, AlexNet	PlantVillage	93.88%
Lu et al. (2017)	DCNN	Self	95.48%
Durmuş et al. (2017)	AlexNet, SqueezeNet	PlantVillage	95.65%
Ferentinos (2018)	CNN	Plantvillage	99.53%
Ma et al. (2018)	DCNN	Self	93.40%
Zhang et al. (2018)	GoogleNet	PlantVillage	98.90%
Rangarajan et al. (2018)	AlexNet, VGG16	PlantVillage	97.49%
Hu et al. (2019)	VGG16	Self	90.00%
Too et al. (2019)	DenseNets	PlantVillage	99.75%
Singh et al. (2019)	Multilayer-CNN	PlantVillage, Self	97.13%
Kamal et al. (2019)	Reduced MobileNet	PlantVillage	98.34%
Geetharamani and Pandian (2019)	CNN	PlantVillage	96.46%
Huynh et al. (2020)	CNN	Flavia, Swedish	98.22%
Chen et al. (2020)	VGGNet, Inception	PlantVillage, Self	92.00%
This study	Hybrid-CNN	PlantVillage	99.27%

3 Materials and methods

The recent studies on image classification tasks in the literature primarily utilizes deep learning methods, such as CNN, and the results have been promising. The advantage of CNNs is that they can learn intended features from training data that automatically consists of both local and deep patterns instead of relying on manual feature extraction (Sharif et al. 2019).

CNNs consist of four main layers: a convolution layer, an activation function layer, a pooling layer, and a fully connected layer. A convolution layer is used to extract various features by convoluting the input data. Hence, the convolution operation is the most important component of CNN. Convolution layers consist of several filters (also known as kernels) that are used to calculate different feature maps. The feature maps are obtained by applying the evolutionary layers several times, usually depending on the size of the input image. The pooling layer's purpose is to reduce the spatial size of the feature map and computation. The pooling operation is applied after convolution, such that the output of the convolution layer serves as the input of the pooling layer. The feature maps obtained after completion of the convolution layers are then passed through an activation unit. The activation functions, which introduce nonlinearities to the CNN, are preferred for detecting nonlinear features in multi-layer networks (Gu et al. 2018). There is some kind of activation functions, such as Hyperbolic Tangent, Sigmoid, and ReLU (Rectified Linear Unit); however, the nonlinear ReLU activation function is more effective and is more frequently used in deep learning studies (Zhang et al. 2019b). After completion of the convolution and pooling operations, the feature map is converted to a one-dimensional vector, which is fed as an input into the fully connected layer. This feature vector is classified and predicted in the output of the network using the fully connected layer. In Fig. 1, an example standard CNN architecture with two convolution and pooling layers is presented.

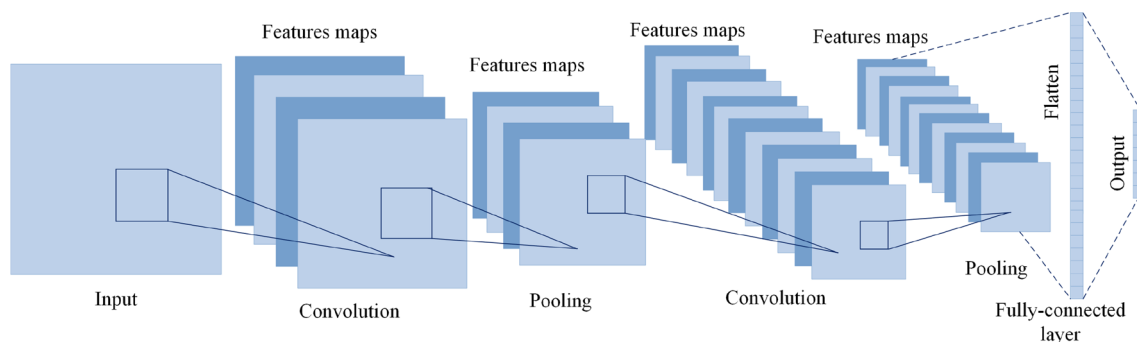


Fig. 1 Standard convolutional neural network

3.1 Depthwise separable convolution

Depthwise separable convolution has been used by Xception (Chollet 2017) and MobileNet (Howard et al. 2017) architectures. This study, inspired by the performance advantage of Xception and MobileNet, uses a CNN model using depthwise separable convolution. While standard convolution is performed in a single step and involves the application of filters into all input channels as well as the combination of these values, depthwise separable convolution contains two different layers: depthwise convolution (which performs the filtering step) and pointwise convolution (which performs the combining step). Figure 2 shows standard convolutions and depthwise separable convolutions, separately. As shown in the figure, a depthwise convolution applies only a single filter to each input channel separately, while pointwise convolution applies a 1×1 convolution that combines different channels to obtain new features (Chollet 2017).

The computational cost of standard convolutions is computed as:

$$D_K^2 \times M \times N \times D_F^2 \quad (1)$$

where D_F is the width and height of a square input feature map, D_K is the dimension of the filter assumed to be square, M is the number of the input channels, and N is the number of filters which means output channels.

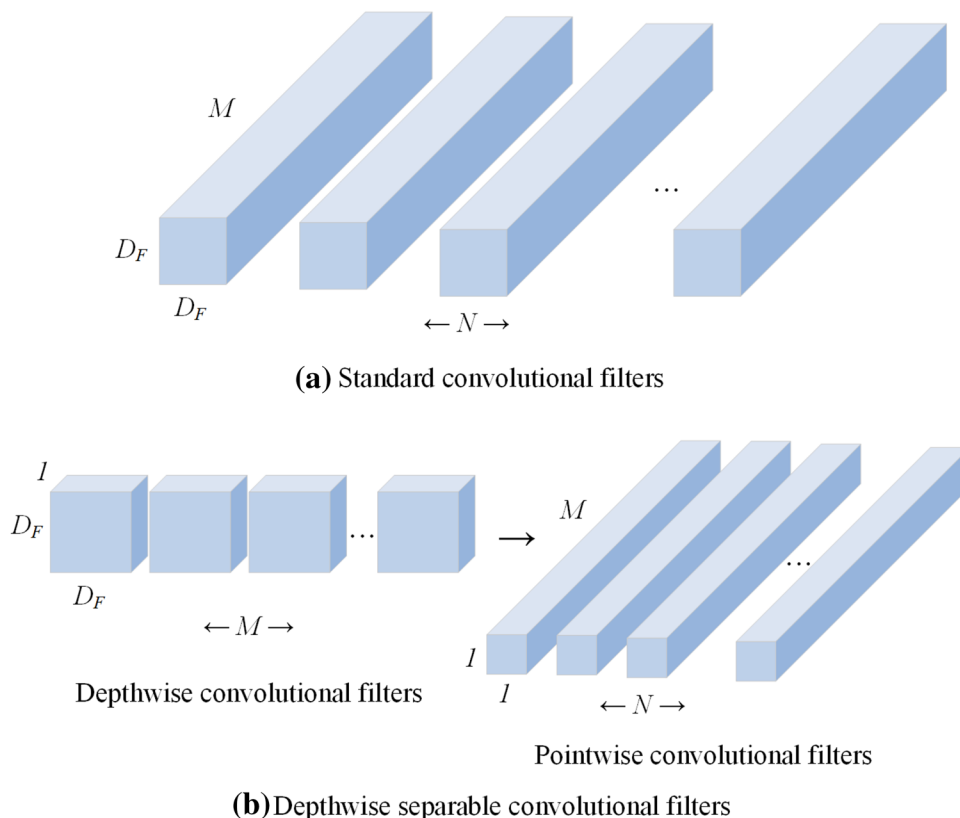
The computational cost of depthwise separable convolutions, which is the sum of the depthwise and pointwise convolutions, computed as:

$$D_K^2 \times M \times D_F^2 + M \times N \times D_F^2 \quad (2)$$

The ratio of the computational cost of the standard convolutions to the depthwise separable convolution is:

$$\frac{D_K^2 \times M \times N \times D_F^2}{D_K^2 \times M \times D_F^2 + M \times N \times D_F^2} = \frac{1}{N} + \frac{1}{D_K^2} \quad (3)$$

Fig. 2 **a** The standard convolutional filters, **b** depthwise separable convolutional filters



It can be understood from the Eq. (3) that the computational complexity of the depthwise separable convolution is reduced to about the square of the number of filters of the standard convolution. As the size of the convolution filters increases, the number of parameters used in the standard convolution increases much more than those used in the depthwise separable convolution.

3.2 The proposed hybrid CNN architecture

Deep learning methods are a subset of machine learning methods. The fully connected layer of a deep learning model is an artificial neural network with a big set of hidden layers. The machine learning methods are fed by features such as statistical or signal processing values to construct a representation of data. Deep learning models need more hardware requirements and time-consuming. Nevertheless, recent studies prefer deep learning models with the use of cloud systems and GPUs.

Although the literature features many CNN approaches for classification and detection, researchers put effort to create different variants of CNN architectures to increase accuracy or reduce the numbers of parameters and computation cost. The training phase of the deep learning algorithm is carried out on powerful computers but the prediction phase is carried out on end devices. Those end devices such as mobile phones or other embedded device solutions

are limited for loading and running complex models, they need lighter models to deploy and run. Considering all these constraints, in this study, a lighter CNN model has been proposed with fewer parameters and faster compared to standard CNNs.

The Inception architecture is designed to perform well even under high computational efficiency and a low number of parameters. It concatenates the filter outputs from various filter sizes, and besides, it also provides a dimensionality reduction. As shown in Fig. 3, the original Inception architecture (Szegedy et al. 2015) consists of four parallel layers. The three layers consist of convolutions with sizes of 1×1 , 3×3 , and 5×5 to extract information from different spatial sizes. The other layer consists of a 3×3 maximum pooling, followed by a 1×1 convolutions. 1×1 convolutions follow 3×3 and 5×5 convolutions to reduce the number of input channels, reducing the complexity of the model.

Instead of sequential convolution operations in standard CNN approaches, Inception architecture can extract better features by performing convolution and pooling operations in parallel. However, the existing Inception module can be optimized in terms of parameters without reducing the success rate in feature extraction. The depthwise separable convolution factorizes the standard convolution into a depthwise convolution together with a pointwise convolution. The depthwise convolution is used to independently perform a spatial convolution for each channel of the input

Fig. 3 Inception architecture

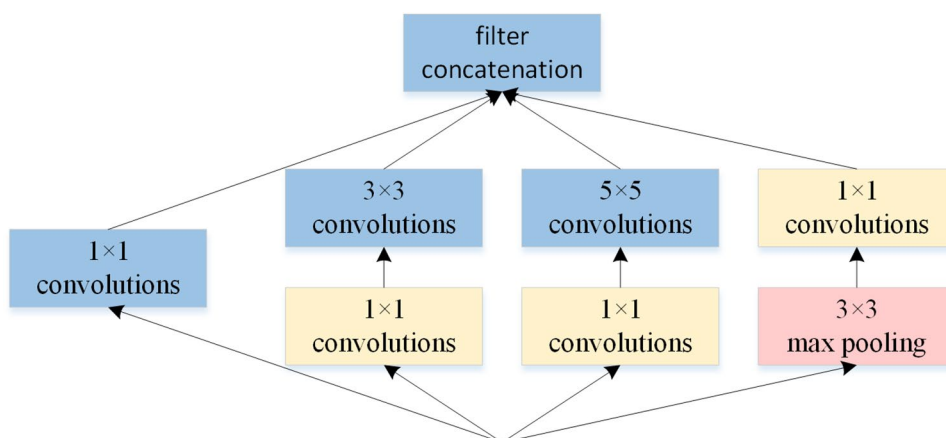
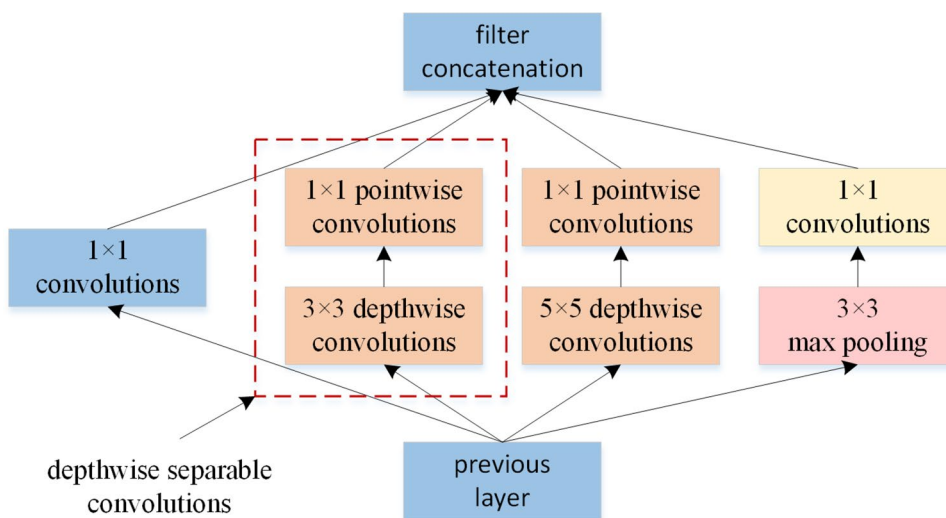


Fig. 4 Modified Inception architecture with depthwise and pointwise convolutions



image, pointwise convolution then is used to combine the output obtained from the depthwise convolution. Depthwise separable convolution separates feature extraction and combination, thereby reducing the number of parameters and redundant computation cost (Howard et al. 2017). By combining different advantages of Inception architecture and depthwise separable convolutions, a hybrid model with a high success rate and a low number of parameters is aimed.

In the study, a hybrid CNN approach has been proposed by combining the Inception architecture with the depthwise separable convolutions. Thus, a new model with low computation cost has been developed due to the use of fewer parameters. In the proposed approach, as shown in Fig. 4, the sequential 1×1 and 3×3 , 1×1 and 5×5 convolutions in the standard Inception architecture have been replaced with 3×3 depthwise and 1×1 pointwise convolutions, and 5×5 depthwise and 1×1 pointwise convolutions. The number of parameters of the Inception with standard convolutions and proposed architecture is given in Tables 2 and 3, respectively. The total number of parameters calculated with Eqs.

Table 2 Parameters of the inception architecture

Type	Input	Filter (size/number)	Parameters
Input image	$256 \times 256 \times 3$	–	–
Conv. 1	Input image	$1 \times 1 / 32$	128
Conv. 2	Input image	$1 \times 1 / 32$	128
Conv. 3	Conv. 2	$3 \times 3 / 32$	9248
Conv. 4	Input image	$1 \times 1 / 32$	128
Conv. 5	Conv. 4	$5 \times 5 / 32$	25,632
Max-pooling	Input image	$3 \times 3 / 32$	–
Conv. 6	Max-pooling	$1 \times 1 / 32$	128
			35,392

(1) and (2) is 35,392 for the Inception architecture and 614 for the proposed architecture. With this change, the number of parameters used for calculation has been reduced by 58 times with extremely successful results. The effect of parameter reduction on computation cost is dramatic on the whole hybrid CNN architecture in the study. The proposed

Table 3 Parameters of the modified inception architecture with depthwise separable convolutions

Type	Input	Filter (size/number)	Parameters
Input image	$256 \times 256 \times 3$	-	-
Conv. 1	Input image	$1 \times 1 / 32$	128
Depthwise conv. 2	Input image	$1 \times 1 / 32$	155
Pointwise conv. 2	Depthwise conv. 2	$3 \times 3 / 32$	
Depthwise conv. 3	Input image	$1 \times 1 / 32$	203
Pointwise conv. 3	Depthwise conv. 3	$5 \times 5 / 32$	
Max-pooling	Input image	$3 \times 3 / 32$	-
Conv. 4	Max-pooling	$1 \times 1 / 32$	128
			614

hybrid CNN architecture does not consist entirely of modified Inception architecture. Besides, there are also sequential depthwise separable convolution layers like standard convolutions.

Figure 5 shows the proposed hybrid CNN architecture used in the study. As shown in the figure, modified Inception architecture with depthwise separable convolutions and consecutive depthwise separable convolutions and pooling operations have been carried out. Modified Inception architecture has been applied two times. Apart from the modified Inception blocks, four depthwise separable convolution layers and four pooling layers have been applied consecutively. After all convolution and pooling

operations were completed, a fully connected (FC) layer and softmax classifier have been applied in the model. The ReLU activation function has been applied to all convolution layers.

While the total number of parameters in the proposed model is 76,576, the number of parameters required is 303,546 if it is realized with standard CNN. Thus, about 75% reduction in the number of parameters has been achieved. The number of parameters varies according to the number and size of the filters used for convolution operations in the CNN design. In this respect, as the number of convolution processes, filter sizes, and numbers in the models' increases, the number of parameters in the standard CNN increases more than the number of parameters in the proposed hybrid model. All the operations applied to the proposed architecture and the standard CNN and the required number of parameters are given in Table 4. DS-Conv term in the table refers to depthwise separable convolution.

Dropout has been applied to the fully connected layer. One of the reasons for poor performance in deep learning is overfitting, a solution for the reduction of which is dropout (Srivastava et al. 2014), a simple and powerful regularization technique used in deep learning models. The basic approach in dropout is to randomly and temporarily drop units in layers during the training. Besides, Batch Normalization has been also used to improve the speed and performance of the model. It normalizes the output of the previous layer by the batch mean and variance. Thus, the number of training epochs required to train the CNN model is decreased significantly. In the model, the Batch

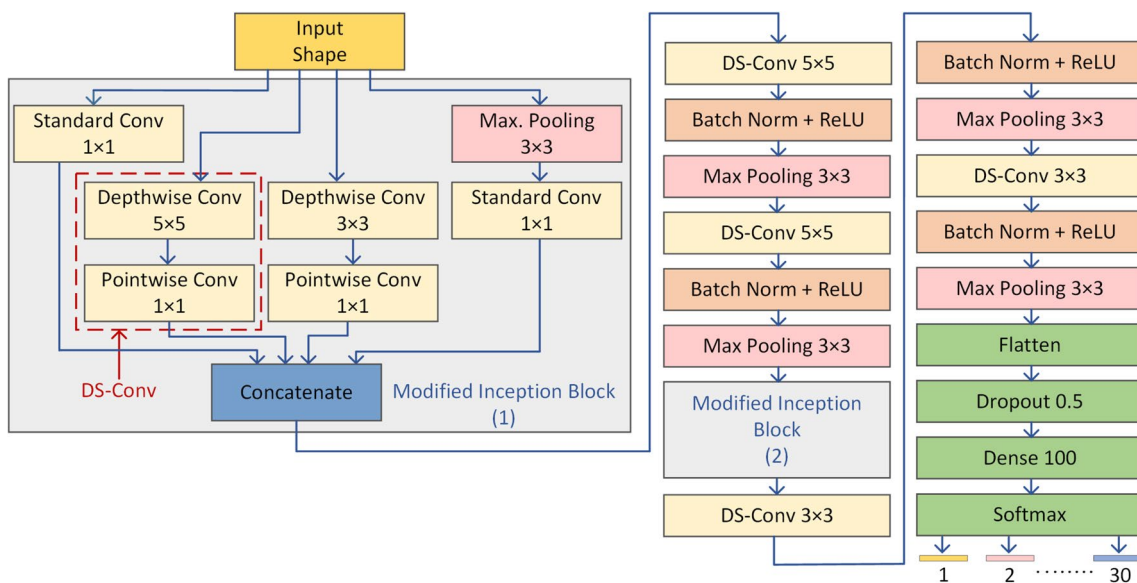


Fig. 5 The proposed hybrid CNN architecture

Table 4 Parameters of the standard CNN and the proposed hybrid CNN architectures

Standard CNN architecture			Proposed hybrid architecture		
Type	Output size	Parameters	Type	Output size	Parameters
Input image	$256 \times 256 \times 3$	–	Input image	$256 \times 256 \times 3$	–
Inception	$256 \times 256 \times 32$	35,392	Modified inception	$256 \times 256 \times 32$	614
Conv (5,5)	$126 \times 126 \times 32$	102,432	DS-conv (5,5)	$126 \times 126 \times 32$	7328
Batch norm.	$126 \times 126 \times 32$	128	Batch norm.	$126 \times 126 \times 32$	128
Max-pooling (3,3)	$62 \times 62 \times 32$	–	Max-pooling (3,3)	$62 \times 62 \times 32$	–
Conv (5,5)	$58 \times 58 \times 32$	25,632	DS-conv (5,5)	$58 \times 58 \times 32$	1856
Batch norm.	$58 \times 58 \times 32$	128	Batch norm.	$58 \times 58 \times 32$	128
Max-pooling (3,3)	$28 \times 28 \times 32$	–	Max-pooling (3,3)	$28 \times 28 \times 32$	–
Inception	$28 \times 28 \times 32$	39,104	Modified inception	$28 \times 28 \times 32$	5312
Conv (3,3)	$26 \times 26 \times 32$	36,896	DS-conv (3,3)	$26 \times 26 \times 32$	5280
Batch norm.	$26 \times 26 \times 32$	128	Batch norm.	$26 \times 26 \times 32$	128
Max-pooling (3,3)	$12 \times 12 \times 32$	–	Max-pooling (3,3)	$12 \times 12 \times 32$	–
Conv (3,3)	$10 \times 10 \times 32$	9248	DS-conv (3,3)	$10 \times 10 \times 32$	1344
Batch norm.	$10 \times 10 \times 32$	128	Batch norm.	$10 \times 10 \times 32$	128
Max-pooling (3,3)	$4 \times 4 \times 32$	–	Max-pooling (3,3)	$4 \times 4 \times 32$	–
Flatten	$1 \times 1 \times 512$	–	Flatten	$1 \times 1 \times 512$	–
Dropout	0.5	–	Dropout	0.5	–
Fully-connected	$1 \times 1 \times 100$	51,300	Fully-connected	$1 \times 1 \times 100$	51,300
Softmax	$1 \times 1 \times 30$	3030	Softmax	$1 \times 1 \times 30$	3030
		303,546			76,576

Normalization process has been applied after all convolution layers except the modified Inception blocks.

4 Experimental results

The experiments have been conducted on plant leaf images to evaluate the performance of the proposed hybrid CNN model. A total of 50,136 different leaf images of 30 classes from 14 different leaves, with healthy and diseased ones, obtained from the PlantVillage dataset, have been used for training and testing of the proposed model. Figure 6 shows sample images of different leaves from the PlantVillage dataset used in the study. All images in the dataset has been resized to 256×256 pixels in the pre-processing stage. The experiments have been implemented in Python with a single NVIDIA Geforce GTX-1080 GPU. In the experiments, the networks have been trained on the training set for 150 epochs and the averages results of the experiments have been taken. Adam optimizer with a batch-size of 32 has been used.

One of the most important requirements for the training of the model in deep learning applications is the sufficient number of data in the dataset. Because having a large dataset directly affects the performance of the model. Some of the studies have used data augmentation techniques to artificially expand the number of images used in training.



Fig. 6 Images from the diseased and healthy classes of leaves

Although the dataset used in this study has enough number of images for some classes, there are few for some. Data augmentation methods (Shorten and Khoshgoftaar 2019) in which the training images are flipping horizontally or

vertically, and rotating images right or left on an axis from 1° to 359°, and shifting images up, down, left or right have been used in the study to boost the number of images for classes with fewer images.

A *k*-fold cross-validation strategy with *k*=4 has been performed for further analysis of the classification performance. The entire dataset has been randomly divided by 4 subsets. One of these subsets has been used for testing and the other 3 subsets have been used for training. Although the dataset is large, considering 30 different classes, the number of images per class is small. The more the number of *k*-fold cross-validation is chosen, the less the number of test images used after each training. In this regard, the *k* number of cross-validation has been taken as 4 in the study. The increase in the test data reveals the performance of the model more. The validation accuracy of the proposed approach reaches the best 99.27% and an average of 99%. The test procedure of the proposed model has been repeated 4 times with different subsets. With 4-fold, 75% of the dataset has been used as training and 25% as test data at every different training phase. As can be seen from the results of performance metrics given in Fig. 7, although the proposed model uses very few parameters, it succeeds like the other methods which are experimented on the PlantVillage dataset. A confusion matrix has been created to calculate the performance of the model. Accuracy and loss graphics with 4-fold cross-validation for training and validation of the proposed architecture is shown in Fig. 8, respectively.

Furthermore, the receiver operating characteristic (ROC) curve and the area under the curve (AUC) have been used in the testing dataset for 30 classes as given in Fig. 9. ROC curve is one of the popular metrics that shows the

performance of a model graphically. AUC values are in a range [0,1], and a larger AUC specifies better performance. An AUC value of 1 indicates that the prediction is 100% correct. The result from the ROC curve shows that the proposed hybrid model reached a high performance with a micro average AUC value of 0.994 and a macro average AUC value of 0.993.

5 Conclusion

The major contribution of this study is to present a deep learning model that requires less computation cost and time for embedded and mobile devices with limited resources. To do this, a new hybrid model by combining the Inception architecture and depthwise separable convolutions has been proposed. The model reduces the number of parameters by approximately 75%; therefore, it operates faster than the standard CNN. With the reduction of the number of parameters of the hybrid model, it is clearly seen that the computation cost and time decrease compared to the standard CNN. The training and testing for detection have been carried out on the PlantVillage dataset of 50,136 images containing 30 classes from 14 different leaves, including healthy and diseased ones. The experimental results demonstrate that the proposed approach is also effective and has a high level of detection accuracy. The average accuracy of the proposed hybrid model is the best 99.27% and an average of 99% with a 4-fold cross-validation process. In addition to accuracy, the performance of the model has been evaluated in terms of precision, recall, f1 score metrics, and ROC curves. The

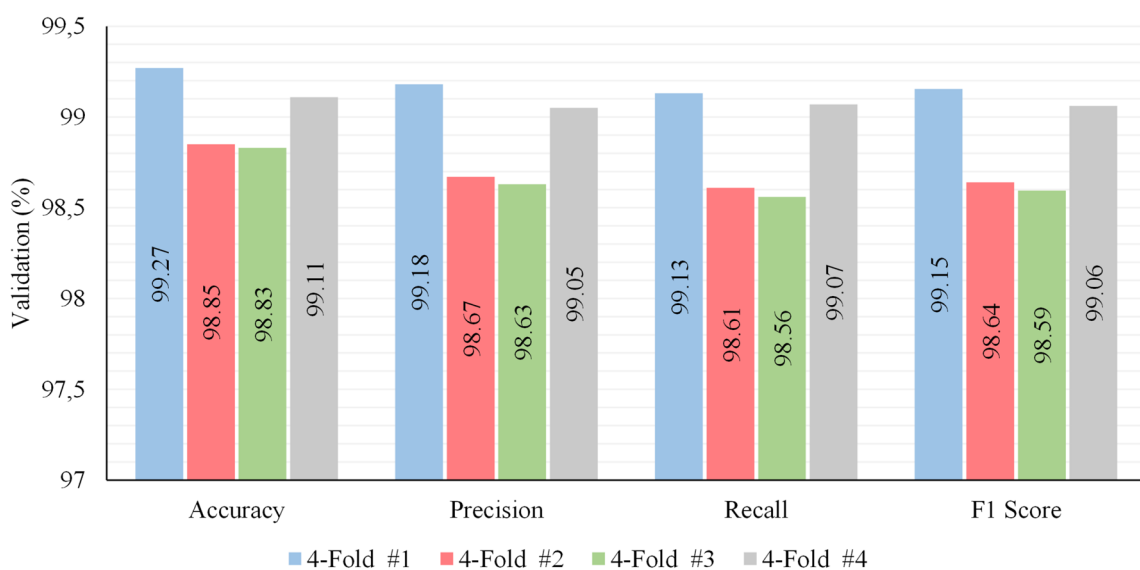


Fig. 7 The performance metric results of the proposed hybrid model with 4-fold cross-validation

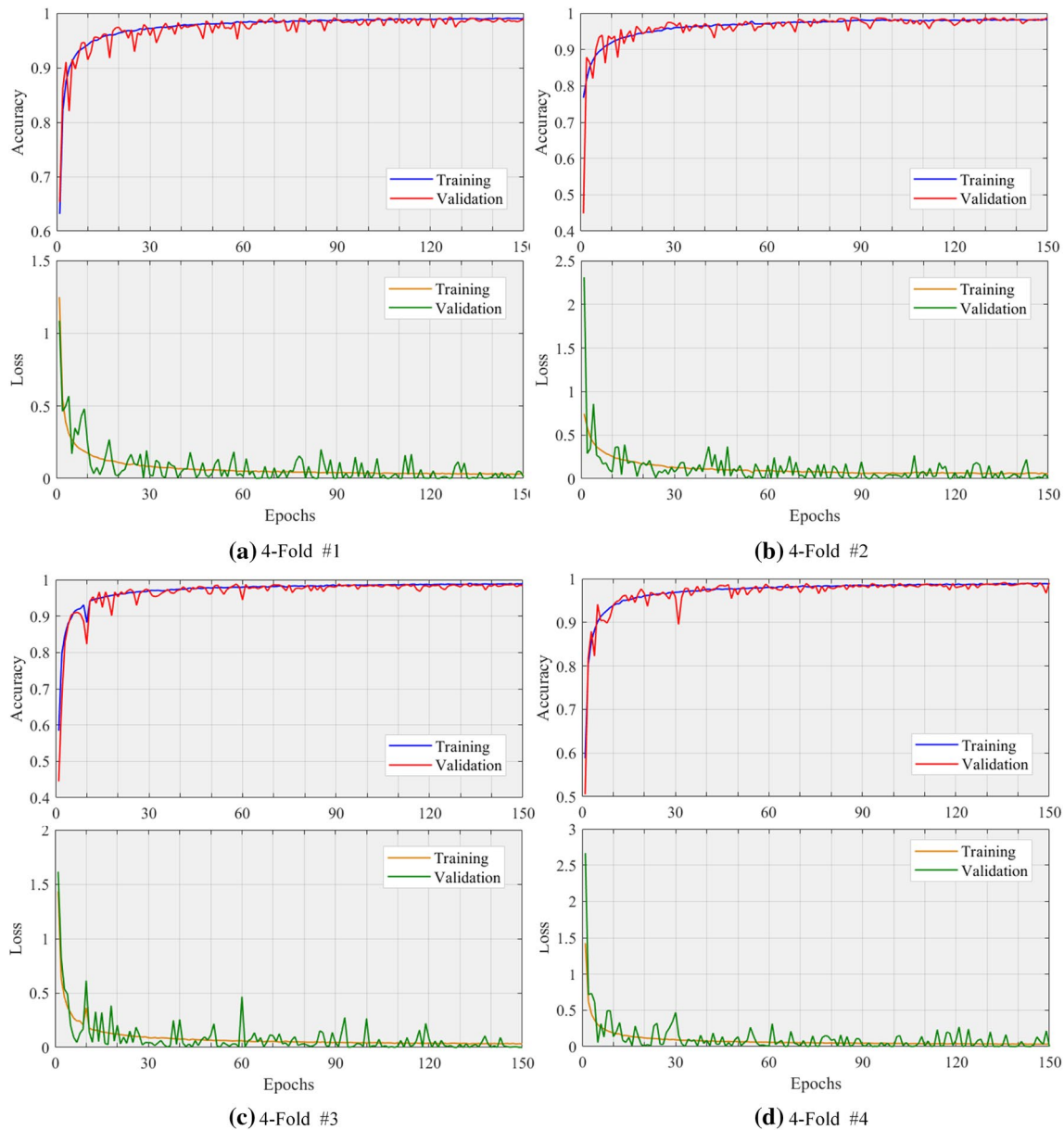


Fig. 8 Accuracy and loss graphics with 4-fold cross-validation of the proposed hybrid model

average values of these metrics are 98.88%, 98.84%, and 98.86%, respectively, which shows a good prediction capability of the proposed method. The decrease in the number of parameters and the increasing speed of operation show

that the proposed approach can be used effectively to realize real-time object detection applications, especially in mobile devices with limited resources.

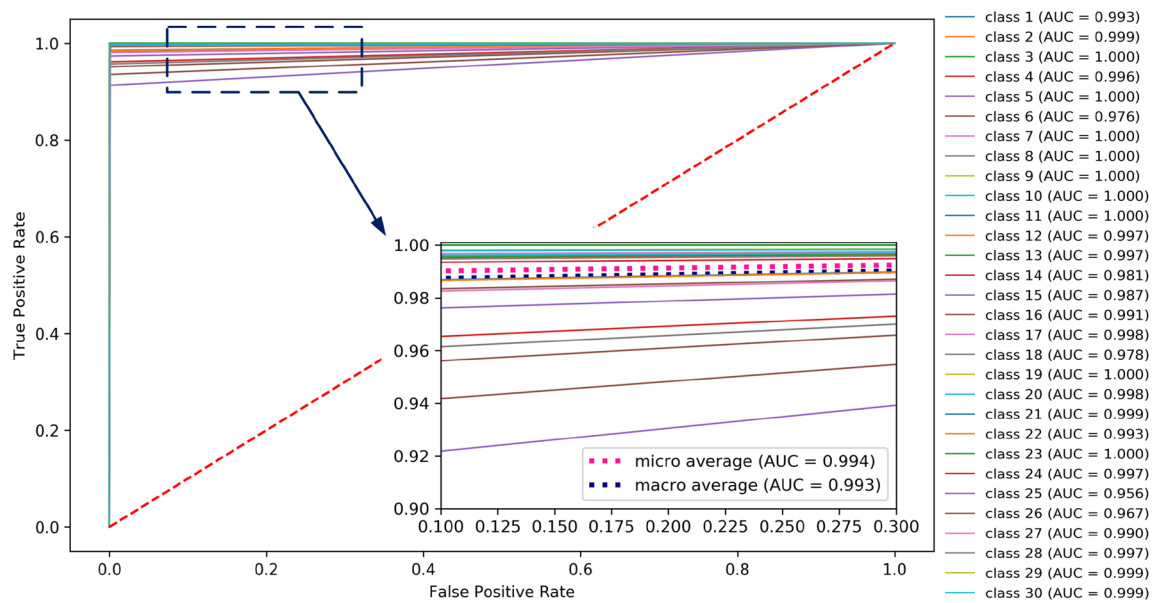


Fig. 9 Validation ROC Curve on plant leaf dataset for 30 classes

References

Barbedo JGA (2016) A review on the main challenges in automatic plant disease identification based on visible range images. *Biosyst Eng* 144:52–60. <https://doi.org/10.1016/j.biosystemseng.2016.01.017>

Chen J, Chen J, Zhang D, Sun Y, Nanekaran Y (2020) Using deep transfer learning for image-based plant disease identification. *Comput Electron Agric* 173:105393. <https://doi.org/10.1016/j.compag.2020.105393>

Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1251–1258. <https://doi.org/10.1109/CVPR.2017.195>

Durmuş H, Güneş EO, Kırıcı M (2017) Disease detection on the leaves of the tomato plants by using deep learning. In: *2017 6th international conference on agro-geoinformatics, IEEE*, pp 1–5. <https://doi.org/10.1109/Agro-Geoinformatics.2017.8047016>

Ferentinos KP (2018) Deep learning models for plant disease detection and diagnosis. *Comput Electron Agric* 145:311–318. <https://doi.org/10.1016/j.compag.2018.01.009>

Geetharamani G, Pandian A (2019) Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput Electr Eng* 76:323–338. <https://doi.org/10.1016/j.compeleceng.2019.04.011>

Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J et al (2018) Recent advances in convolutional neural networks. *Pattern Recogniti* 77:354–377. <https://doi.org/10.1016/j.patcog.2017.10.013>

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. <https://doi.org/10.1109/CVPR.2016.90>

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:170404861*

Hu G, Wu H, Zhang Y, Wan M (2019) A low shot learning method for tea leaf’s disease identification. *Comput Electron Agric* 163:104852. <https://doi.org/10.1016/j.compag.2019.104852>

Hughes D, Salathé M (2015) An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:151108060*

Huynh HX, Truong BQ, Thanh KTN, Truong DQ (2020) Plant identification using new architecture convolutional neural networks combine with replacing the red of color channel image by vein morphology leaf. *Vietnam J Comput Sci* 7(02):197–208. <https://doi.org/10.1142/S2196888820500116>

Jin R, Lu L, Lee J, Usman A (2019) Multi-representational convolutional neural networks for text classification. *Comput Intell* 35(3):599–609. <https://doi.org/10.1111/cojin.12225>

Kamal K, Yin Z, Wu M, Wu Z (2019) Depthwise separable convolution architectures for plant disease classification. *Comput Electron Agric* 165:104948. <https://doi.org/10.1016/j.compag.2019.104948>

Kamilaris A, Prenafeta-Boldú FX (2018) Deep learning in agriculture: a survey. *Comput Electron Agric* 147:70–90. <https://doi.org/10.1016/j.compag.2018.02.016>

Kaya A, Keceli AS, Catal C, Yalic HY, Temucin H, Tekinerdogan B (2019) Analysis of transfer learning for deep neural network based plant classification models. *Comput Electron Agric* 158:20–29. <https://doi.org/10.1016/j.compag.2019.01.041>

Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - (NIPS’12)*, vol 1. Curran Associates Inc., Red Hook, NY, USA, pp 1097–1105

Lee SH, Chan CS, Mayo SJ, Remagnino P (2017) How deep learning extracts and learns leaf features for plant classification. *Pattern Recogniti* 71:1–13. <https://doi.org/10.1016/j.patcog.2017.05.015>

Li Y, Wang Y, Li D (2019) Privacy-preserving lightweight face recognition. *Neurocomputing* 363:212–222. <https://doi.org/10.1016/j.neucom.2019.07.039>

- Lu Y, Yi S, Zeng N, Liu Y, Zhang Y (2017) Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* 267:378–384. <https://doi.org/10.1016/j.neucom.2017.06.023>
- 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. <https://doi.org/10.1016/j.compag.2018.08.048>
- Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. *Front Plant Sci* 7:1419. <https://doi.org/10.3389/fpls.2016.01419>
- Pawara P, Okafor E, Surinta O, Schomaker L, Wiering M (2017) Comparing local descriptors and bags of visual words to deep convolutional neural networks for plant recognition. In: international conference on pattern recognition applications and methods (ICPRAM), pp 479–486
- Peng Y, Wang Y (2019) Real-time forest smoke detection using hand-designed features and deep learning. *Comput Electron Agric* 167:105029. <https://doi.org/10.1016/j.compag.2019.105029>
- Rangarajan AK, Purushothaman R, Ramesh A (2018) Tomato crop disease classification using pre-trained deep learning algorithm. *Proc Comput Sci* 133:1040–1047. <https://doi.org/10.1016/j.procs.2018.07.070>
- Rangarajan Aravind K, Raja P (2020) Automated disease classification in (selected) agricultural crops using transfer learning. *Automatika* 61(2):260–272. <https://doi.org/10.1080/00051144.2020.1728911>
- Sardogan M, Tuncer A, Ozen Y (2018) Plant leaf disease detection and classification based on cnn with lvq algorithm. In: 2018 3rd international conference on computer science and engineering (UBMK), IEEE, pp 382–385. <https://doi.org/10.1109/UBMK.2018.8566635>
- Sharif M, Attique Khan M, Rashid M, Yasmin M, Afza F, Tanik UJ (2019) Deep cnn and geometric features-based gastrointestinal tract diseases detection and classification from wireless capsule endoscopy images. *J Exp Theor Artif Intell* 1–23. <https://doi.org/10.1080/0952813X.2019.1572657>
- Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. *J Big Data* 6(1):60. <https://doi.org/10.1186/s40537-019-0197-0>
- Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
- 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. <https://doi.org/10.1109/ACCESS.2019.2907383>
- Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(1):1929–1958
- Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9. <https://doi.org/10.1109/CVPR.2015.7298594>
- 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. <https://doi.org/10.1016/j.compag.2018.03.032>
- Zhang K, Cheng K, Li J, Peng Y (2019a) A channel pruning algorithm based on depth-wise separable convolution unit. *IEEE Access* 7:173294–173309. <https://doi.org/10.1109/ACCESS.2019.2956976>
- Zhang S, Huang W, Zhang C (2019b) Three-channel convolutional neural networks for vegetable leaf disease recognition. *Cognit Syst Res* 53:31–41. <https://doi.org/10.1016/j.cogsys.2018.04.006>
- Zhang X, Qiao Y, Meng F, Fan C, Zhang M (2018) Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access* 6:30370–30377. <https://doi.org/10.1109/ACCESS.2018.2844405>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.