


RESEARCH

Open Access



Plant diseases detection with low resolution data using nested skip connections

Hilman F. Pardede^{1*} , Endang Suryawati¹, Vicky Zilvan¹, Ade Ramdan¹, R. Budiarianto S. Kusumo¹, Ana Heryana¹, R. Sandra Yuwana¹, Dikdik Krisnandi¹, Agus Subekti², Fani Fauziah³ and Vitria P. Rahadi³

*Correspondence:
hilm001@lipi.go.id

¹ Reserch Center
for Informatics, Indonesian
Institute of Sciences,
Bandung, Indonesia
Full list of author information
is available at the end of the
article

Abstract

At the moment, there are increasing trends of using deep learning for plant diseases detection. However, their implementations may be difficult in developing countries due to several reasons. First, existing deep learning models are usually trained with images with adequate resolutions. In developing countries however, with limited internet connection, models that would perform well even when data with low resolution are used are needed. Secondly, the generated models are large. Hence, most deep learning based applications are available on-line. Unfortunately, the trend for new deep learning architectures are either have larger models or require a heavy memory usage. So, models with smaller size would be preferred. In this paper, we evaluate various existing deep learning models for plant diseases detection when low resolution data are used. They are: VGGNet, AlexNet, Resnet, Xception, and MobileNet. Our focus is deep convolutional neural network (DCNN) which is commonly applied for image data. We also propose a new DCNN architecture with two branches of concatenated residual networks. It is well known that the deeper the networks the better performance of DCNN. However, DCNN with very deep networks and large number of training parameters is prone to vanishing gradient problems. One solutions for that is to apply residual networks as branches to DCNN. While it is found that increasing the branch of the networks benefit the performance, larger memory are required to train the networks. So, we apply two concatenated residual networks only. We called it Compact Networks (ComNet). We compare our method other with six popular CNN architectures. We evaluate the performance on the PlantVillage dataset and our own dataset. We collected images of tea leaves which consist of 6 classes: 5 classes of diseases that are commonly found in Indonesia and a healthy class. Our experiments show that our method is generally better than referenced DCNN networks.

Keywords: Deep Learning, Convolutional neural network, Light CNN, Plant diseases detection, Tea diseases detection

Introduction

Natural disasters, climate change[54] and plant diseases[51] are among many factors that threaten the food security. Plant diseases in particular, may cause great loss not only for farmers, but also for global economy. For instance, The International Potato Center (CIP) reports that around 15 % loss of potatoes production[24] is due to late blight diseases only. Globally, plant diseases cause more than 20 % crop loss annually[49]. The

plant diseases are even bigger threats for smallholder farmers due their limited knowledge, resources and financial power to deal with them. This is the case in Indonesia, where smallholder farmers comprise the majority of farmers.

Early detection of plant diseases proves to be effective to reduce the risk of crop failure as farmers may perform some curative and preventive actions to avoid further damages. Detection of plant diseases with naked eye inspection would require human experts. A large team of experts that continuously monitor the condition of the farms would be needed and this would cost greatly especially if the area of the farms are large and disperse. For countries like Indonesia where of farmers are distributed in large areas and separated by seas/islands, the government may have limitation to provide experts especially, especially for remote areas. Therefore, automatic detection for plant diseases is needed.

Some works indicate the diseases of the plants by detecting their level of stressness. This could be detected using various methods, for instance, using hyperspectral and multispectral sensing[5, 13, 36, 62], thermal imaging methods[32], chemical substances[6], and/or molecular and genetic level analysis[34, 35]. However, these approaches would be expensive and require experts to operate, and thus would be unattainable for smallholder farmers. Other studies apply various image processing techniques combined with machine learning for pattern searching of the diseases is one solution to provide easily accessible aids for small farmers. Given enough image data of infected plants, we could train machine learning systems that are able to identify the diseases given the corresponding data. Various machine learning methods such as Support Vector Machines (SVM)[10], k-Nearest neighbor (k-NN), Naive Bayes, or Random Forest[19] are applied. For instance, Spectroscopy of plant tissues is used in[26]. In other study[14], multispectral data are used as features for Neural Networks to detect diseases on cucumber. SVM is used as classifier for Huanglongbing citrus diseases detection with fluorescence images in a reported study[57]. PCA and Linear Discriminant Analysis are used Blast rice diseases in[59]. For a review for the use of spectral data for plant diseases detection could refer to[47]. Unfortunately, spectral data are extracted using expensive equipments[44].

Many efforts have been made to develop machine learning based detectors that work with standard image data. Using such data, the efforts focus on developing hand-engineered features from images, usually based on human knowledge on particular problems using various transformations, such that they are good to discriminate each class for recognition. Several examples of engineered features are Histogram of Oriented Gradients (HOG)[11], Scale-Invariant Feature Transform (SIFT)[33], Speeded Up Robust Features (SURF)[4], and Linear Binary Pattern (LBP)[38]. For examples, k-means clustering with neural network are used in[2]. In[42], combinations of Gabor wavelet transform (GWT) and gray level co-occurrence matrix (GLCM) are used as features with k-NN as the classifier. In other study, SIFT features with principle components analysis (PCA) are used with Learning Vector Quantization (LVQ)[41]. Variant of SVM with GLCM and LBP features are used in[1] for citrus. Unfortunately, these engineered features usually require complex computations and processes[47].

Recent advancement in machine learning, called deep learning (DL), paves way for more accurate classifications given simpler features[30]. DL methods have been used in many machine learning tasks, such as speech recognition[12, 46] natural language

processing[9, 48], and computer vision[56]. In the area of computer vision and object recognition[45], DL is state-of-the art for many applications. Most studies in computer vision and object recognition use the convolutional neural networks (CNN)[31] and their variants, such as AlexNet[29], VGGNet[50], GoogleNet[53], Xception[7], and MobileNet[20], ZFNet[61], and ResNet[18]. As it is reported in[45], it is clear that CNN become the dominant technology for object recognition.

For diseases detection, many studies also have implemented the aforementioned architectures. In [37], GoogleNet and AlexNet are used to detect diseases from 19 plants. Meanwhile, ResNet and VGGNet are used to detect diseases on tomatoes in [16]. AlexNet and VGGNet are used to detect diseases of 25 plants in a reported study [15]. Simplified VGGNet is proposed in [39] for potatoes diseases detection. MobileNet combined with the Single Shot Multibox (SSD) model are used to detect diseases on cassava [43]. These works use variants of convolutional neural networks with decent resolution image data (above 128×128 pixel resolutions). Most studies also have not evaluated the robustness of the methods when tested on non-ideal conditions where the image data may be blurred, have different orientations and/or resolutions than training data. To improve robustness, multicondition training on CNN for tea diseases detection is proposed in [60].

In addition, the DL models are placed in the servers due to their large sizes for the implementations. This is due to the large size of the networks. The targeted image data are then transferred to the server for further processing and classification, and the results are transmitted back to the devices. Most studies usually work with decent quality images (with resolutions of 256×256). However, many developing countries such as Indonesia may still have limited speed for internet connectivity, especially in remote areas where many small farmers live. Therefore, it is better to have the systems that are trained with low resolution images. In addition, the systems must be robust against various image transformations as it is very likely that the images are taken on different conditions as the data training.

Meanwhile, it is well-known that increasing the size of the deep learning networks by increasing the depth of the networks is effective improve the performance of deep learning systems. Later variants of CNN usually come with increasing depth of convolutional layers. But, this causes two drawbacks. First, this may increase computational loads due to the high number of model parameters. Second, very deep networks may prone to overfit especially when the number of training data is limited, and hence adding more layers do not necessarily improve the performance. This is called the gradient vanishing problem[23].

One solution to this problem is to use skip connections. Skip connections are layers that are designed to skip few layers of the networks so the layers could reuse the activations of the skip connections layers during training, and hence avoiding the gradient to vanish. Residual network is an example of skip connections. This is applied in ResNet and Xception where residual networks are added to the output of the convolutional layers [18]. However, information fusion as in Xception may alleviate information passed through to the next layers. Meanwhile, to reduce the number of the parameters, other studies replace convolutional layers with separable convolution [25] as in MobileNet and Xception. With this finding, several newer networks

use more branch of skip connections such as DenseNet [21] and ResNext [58]. But, such networks usually require more memory to train and hence may not applicable for machines with limited resources.

The contributions of this paper are two-folds. First is a new DCNN architectures to significantly reduce the number of the training parameters and to overcome vanishing gradient problems. We called it Compact Networks (ComNet). Our method works by applying two branch of concatenation layers only to minimize the need for more memory for training. The first layer is to carry information from previous layers and the second is concatenated the output of first concatenation layer with small kernels convolutional layers to avoid the gradients to vanish during training. We evaluate the performance and robustness of five major DCNN architectures: VGGNet[50], AlexNet[29], Xception[7], MobileNet[20] and ResNet[18] on low resolution data. Secondly, we develop a dataset of tea diseases that consists of 5 types of diseases that are common in Indonesia and a healthy class. In addition to the evaluation our dataset, we also evaluated them on subset of Plantvillage dataset[37] with reduced resolution due our limited resource for training. We evaluated the methods on 3 plants: apple, corn, and potato, with 11 class labels that were made up by 8 types of diseases and three healthy classes.

The remainder of the paper is organized as follow. We explain all the architectures we used in this study in “[Convolutional neural networks and their variants](#)” section. Our proposed method is explained in “[Proposed method](#)” section in more details. We describe our experimental setup in “[Experimental setup](#)” section and the results are discussed and analysed in “[Results and discussions](#)” section. We conclude the paper in “[Conclusion](#)” section.

Convolutional neural networks and their variants

DL caught attentions of many researchers in the areas of machine learning in recent years. DL systems won numerous competitions in pattern recognition and machine learning[3, 8]. For object recognition tasks such as Large Scale Visual Recognition Challenge (ILSVRC)[45], DL achieves the best performance and outperforms many other conventional methods.

Deep learning technologies are mostly based on artificial neural network (ANN). In ANN, perceptrons are stacked in such way to approximate the relation between the inputs with the outputs (usually the target classes). Therefore, DL can be seen as universal function approximators. By having many hidden layer in the network, DL can model any complex relation using large number of hidden layers, allowing the networks to learn about various abstractions of the data in different layers of the networks. This would allow the networks to learn about the representations of the data by themselves given only the raw information. This is one of the advantages of the DL architectures. Thus, it is unnecessary to design a handcrafted features, which is a common approach when conventional machine learning methods are used.

For object recognition, deep convolutional neural networks (DCNN) and their variations are mainly used for object recognitions. DCNN is built upon stacked convolutional neural networks (CNNs). CNN is a variant of feed-forward network (FFN) where the flow of information has no feedback from the output layer to the previous layers. Like a typical FFN, CNN consists of input, hidden, and output layers with the

hidden layers are typically constructed by convolutional layers, pooling layers, and fully connected layers. The CNN architecture is first proposed in the 80's [17]. In the paper, the structure, which is called Neocognitron, has very similar structure to the current CNN except on how the weights are updated. It is updated in unsupervised manner and pre-wired. Meanwhile in current CNN, the weights are updated with the use of gradient descent based methods [31]. CNN is applied for many tasks in computer visions [56]. Currently, CNN is the leading architecture for image recognition, classification, and detection [29].

In convolutional layers, a convolution operation is applied to the input like standard FFN. The difference is, the nodes in a convolutional layer only connected to a particular inputs unlike FFN where they are connected to all inputs. This is one advantages of CNN over FFN on image data. For image data, the local relation between inputs may need to be emphasized first before the network learn more of the global relations of the images. By doing so, the networks can learn larger area of abstraction and different kind of abstraction when the data pass the higher layers of the networks to obtain different abstraction. Additionally, applying FFN on images would be impractical as it would produce significantly larger number of parameters. This could significantly be reduced by using CNN.

A pooling layer is usually included after the convolutional layers. In the pooling layer, outputs of grouped neuron are combined to produce a single node that would be passed into the following layers. The commonly used combination rule is max-pooling where the output is the maximum values of the grouped neurons. Other approach is average pooling where the average value of the nearby inputs is used as the output.

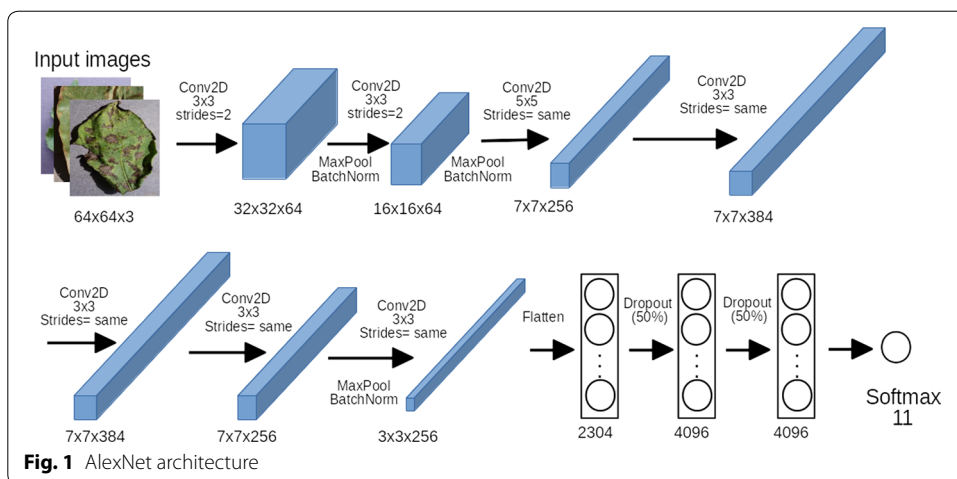
A fully connected layer is usually put at the top of CNN architecture. This layer connect every neuron in previous layer to every neuron in the next layer. So in principle, it is the same with standard multi-layer perceptron network. The purpose is the find the global relation of the data.

There have been many architectures of DCNN. However, due to our limitation for training, we are only capable to evaluate 5 DCNN architectures in this study. They are AlexNet, VGGNet, MobileNet, Xception, and ResNet. Newer architectures such as DenseNet, ResNext, Inception-v4, and Inception-Resnet-v2 [52] are inapplicable in our current machines.

AlexNet

AlexNet is proposed in [29]. AlexNet is the winner of ILSVRC in 2012 where CNN gain its global recognition for the first time. It achieves significantly better performance than conventional methods. It is comprised with five convolutional layers and three fully connected layers. After the first, second, and fifth convolutional layers, max-pooling is applied. Dropout is applied after the first and second fully connected layers.

Originally, AlexNet is used for images with size of 224×224 . To make it able to accommodate low resolution images (64×64), we modify the filter size of each convolutional layers from 96-256-384-256 in the original design to 64-256-384-384-256.



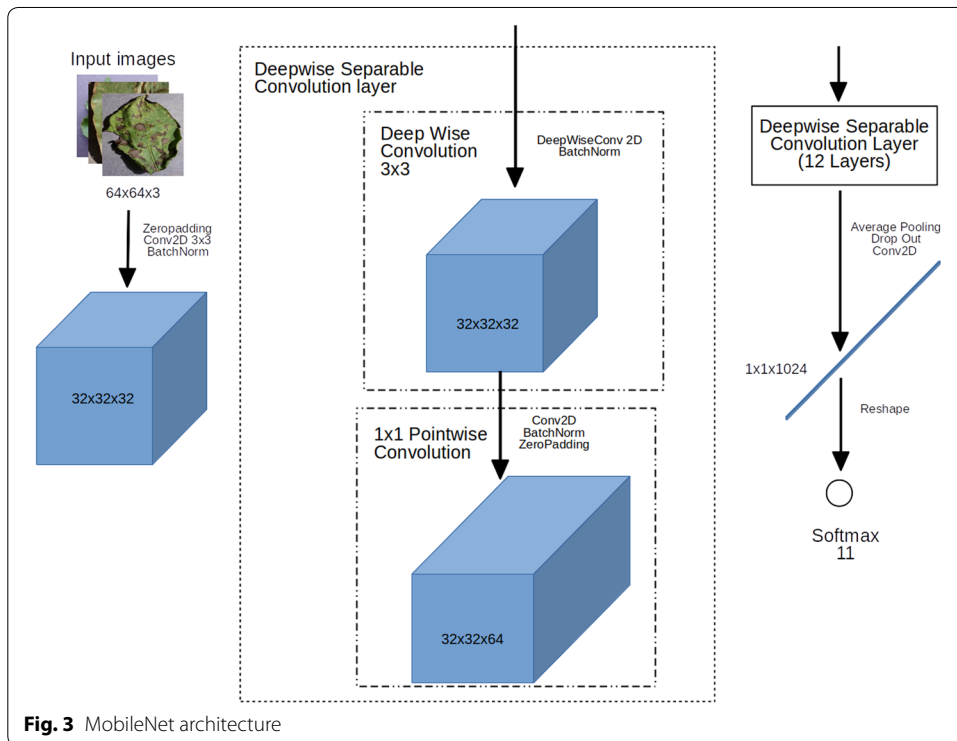
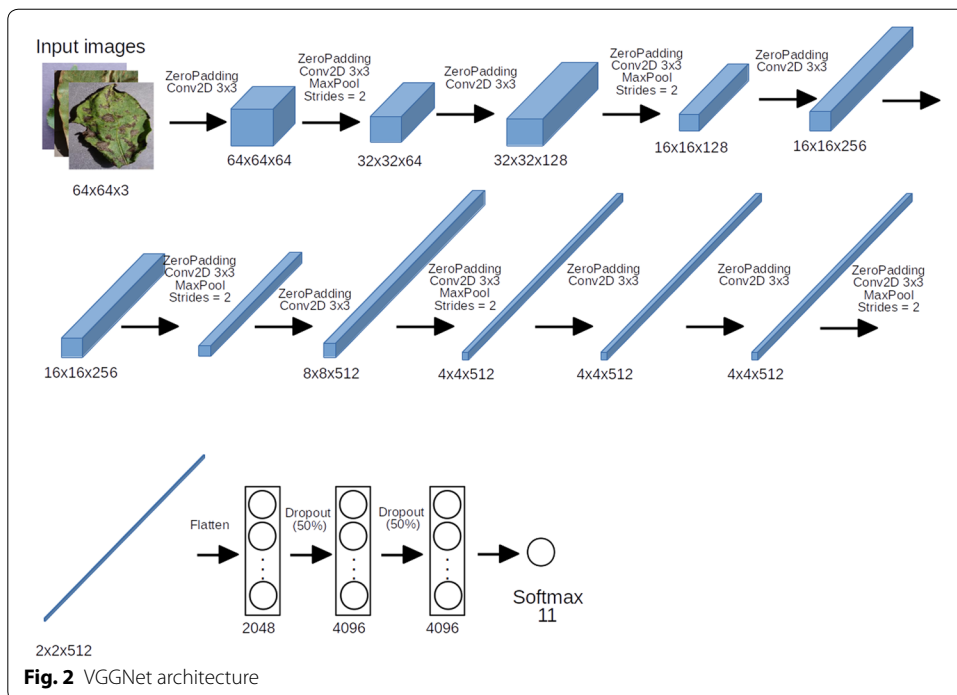
We also modify the kernel size of 11×11 to 3×3 on the first layers. The details of our implementation of AlexNet is shown in Fig. 1.

VGGNet

VGGNet is proposed in [50]. It is the runner-up at the ILSVRC 2014. VGGNet is similar to AlexNet except it is deeper by utilizing smaller convolutional kernels. It has 12 convolutional layers instead of 4 as in AlexNet and 3 fully connected layers. Even though VGGNet achieved the second place in ILSVRC 2014, it is quite popular architecture for learning features from image due to its simplicity. It only uses 3×3 convolution and 2×2 pooling for the whole networks. This architecture shows that improving the networks can be done by keep adding new convolutional layers. One drawback of VGGNet is the size of the networks. It has significantly larger parameters and requires longer training time. There are two variants of VGGNet. They are VGGNet16 with 16 layers and VGGNet19 with 19 layers. Here, we implement VGGNet16. The details of VGGNet we use is shown in Fig. 2.

MobileNet

MobileNet is proposed in [20]. DCNN architectures such as VGGNet require heavy computational loads and great number of parameters. To make it smaller, CNN is replaced by depth-wise separable convolution in MobileNet. Conventional convolutional layers operate by performing summing and filtering operation between input and a defined filter, i.e. cross-correlation operation to be exact, to produce a new set of output. Here, CNN operations are separated by depth-wise separable convolution into two operation: 3×3 depthwise convolution for filtering for the first channel and 1×1 pointwise convolution combining the inputs to other channels. This division produces much smaller computation and model size since full convolution operations are conducted in the first channel only.



The structures of a MobileNet is shown in Fig. 3. First layer uses a regular convolutional layer. Then, it is followed by 13 layers of deep wise convolution and pointwise

convolution layers. After that, average pooling and dropout are used before applying CNN as final layer.

ResNet

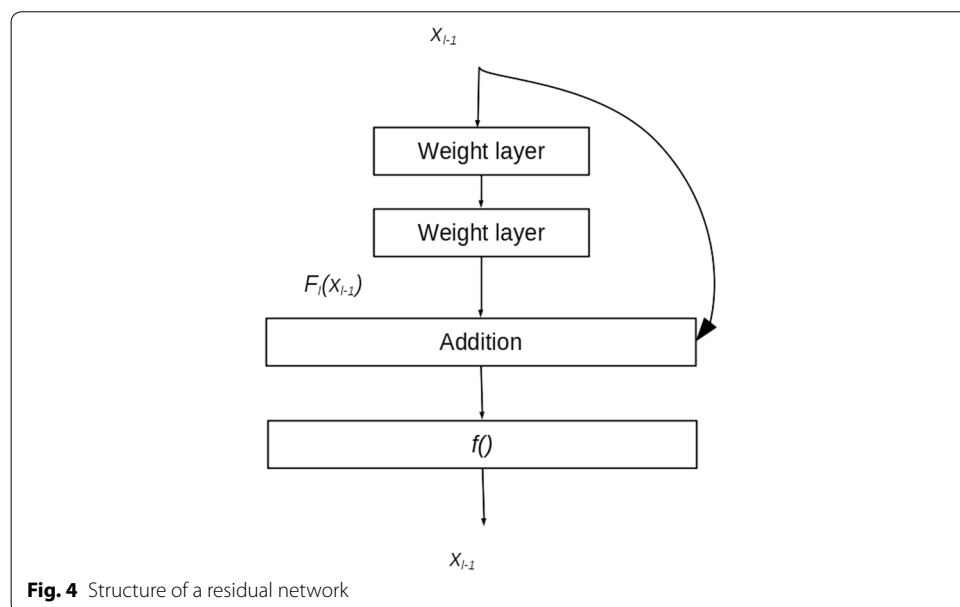
ResNet is proposed to overcome the vanishing gradients problems [55]. Due to the use of small kernels, studies found that VGGNet that use smaller kernels could only works until certain depth. In ResNet, skip connections are implemented. Here, residual networks are introduced as skip connections. The residual networks is illustrated in Fig. 4. The network is designed to learn “the residual” of the networks by adding the input of the block to its output.

In the paper, we implement ResNet50. ResNet is build by stacking Conv Block and Identity block with the structure illustrated in Fig. 5. The image data pass a layers of a CNN in the beginning. ResNet 50 consists of five Conv Blocks and 13 Identity blocks. At final stage, average pooling is applied before passing it into softmax classification.

Xception

Xception is proposed in [7]. Xception stands for eXtreme inception. Xception is similar to ResNet by applying residual networks to enable very deep networks. The difference is Xception apply inception module [53]. Inception modules work by concatenating convolutional layers with various sizes. The purpose of it is to keep good resolution for small area of images while also getting the information on larger area. But, in Xception, the inputs are projected into various output channel by using deepwise separable convolution layer instead. The architecture of Xception is shown in Fig. 6.

In Xception, the data first pass 2 layers of regular CNN before it is passed into the entry block. In entry block, 2 layers of 3×3 Deepwise Separable Convolution are



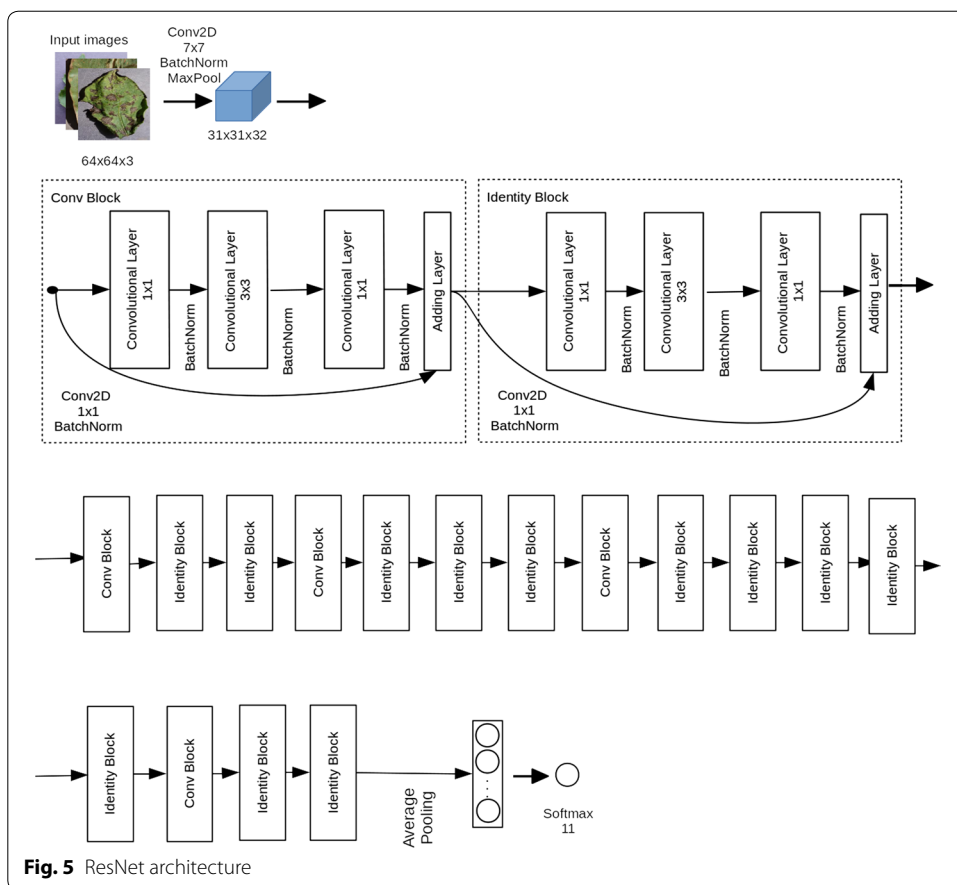


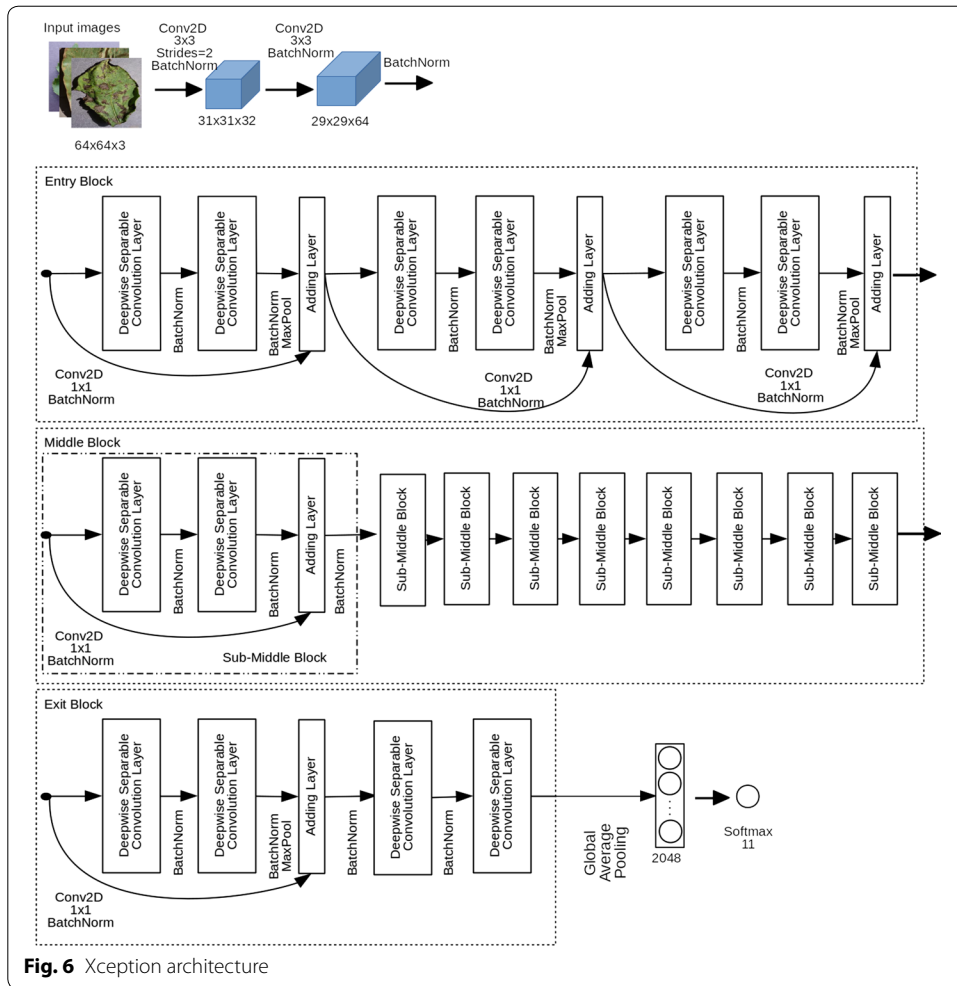
Fig. 5 ResNet architecture

concatenated with 1×1 convolution layer. This procedure is repeated for three times. Then, in middle block, three layers of deepwise separable convolution are stacked. This is repeated for 8 times. Then in exit block, 2 layers of 3×3 Deepwise Separable Convolution are combined with 1×1 convolution layer and it is followed by 2 layers of Deepwise separable convolution.

Proposed method

The most straightforward approach to improve the performance of DCNN is to increase the model complexity by adding the width or the depth of the networks. However, adding more layers would make the networks prone to gradient vanishing/ exploding. As the results, the networks are unable to converge. This could be overcome by adding skip connections. This means a layer skips some layers. By doing so, the networks could reuse activations from much lower layers, and hence vanishing gradient could be avoided. One implementations of skip connections are in residual networks [18]. This concept is applied at Xception, where residual networks are applied as skip connections.

Residual networks work as follows. Let us consider Sub-Middle Block of Xception (See Fig. 6). An output of the l^{th} layer with transformation function f of a residual network can be expressed as:



$$x_l = f(H_l(x_{l-1}) + x_{l-1}) \tag{1}$$

where H_l is the nonlinear transformation due to stacking process. In Xception, it is a composite of batch normalization, rectified linear units (ReLU), pooling, and convolutional layers. In Xception, a layer receives the outputs of previous layers and added them with the residual networks. Therefore, vanishing gradients could be avoided.

With this finding, more recent studies apply more branches of skip connections as in DenseNet[21], ResNext[58], and Inception-V4[52]. However, due to the multiple branches of skip connections, it requires larger memory capacity for the GPU, making it inapplicable for computers with limited GPU memory size.

To overcome this, we limit only two branches of skip connections. In addition, due to the addition process in Xception, some information may be lost. To avoid this, we to concatenate the layers instead. We called it the compact modules since all the input compacted at the output. The comparisons between the proposed architectures and residual networks is illustrated in Fig. 7.

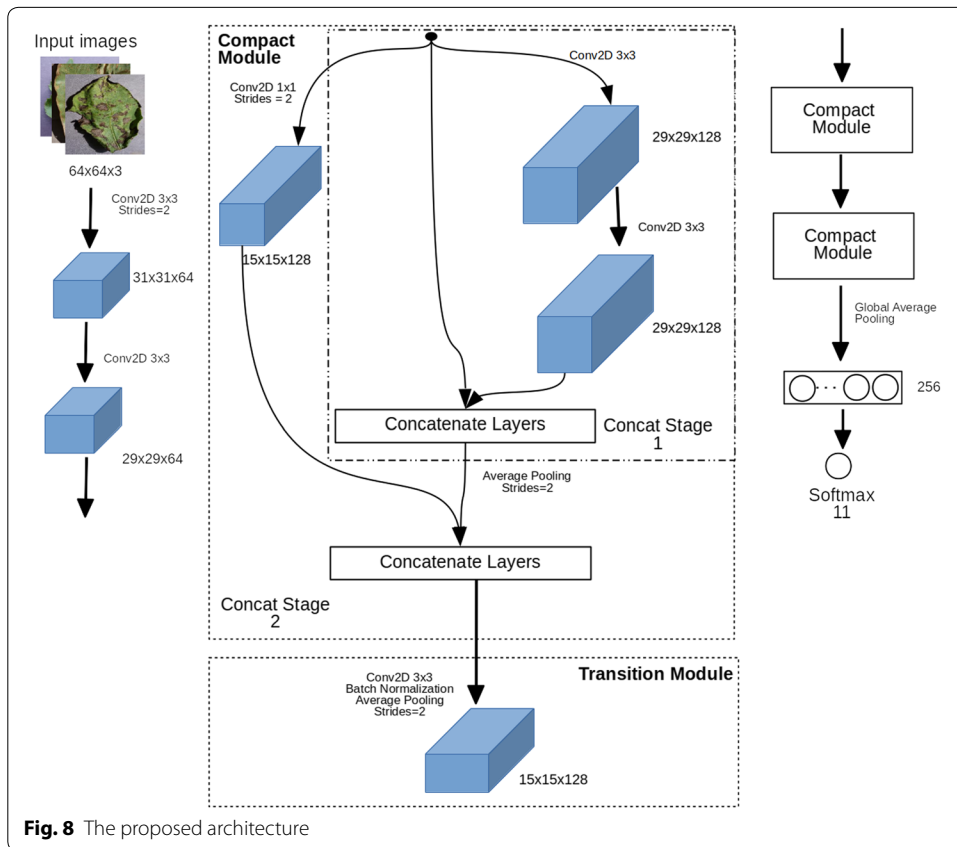
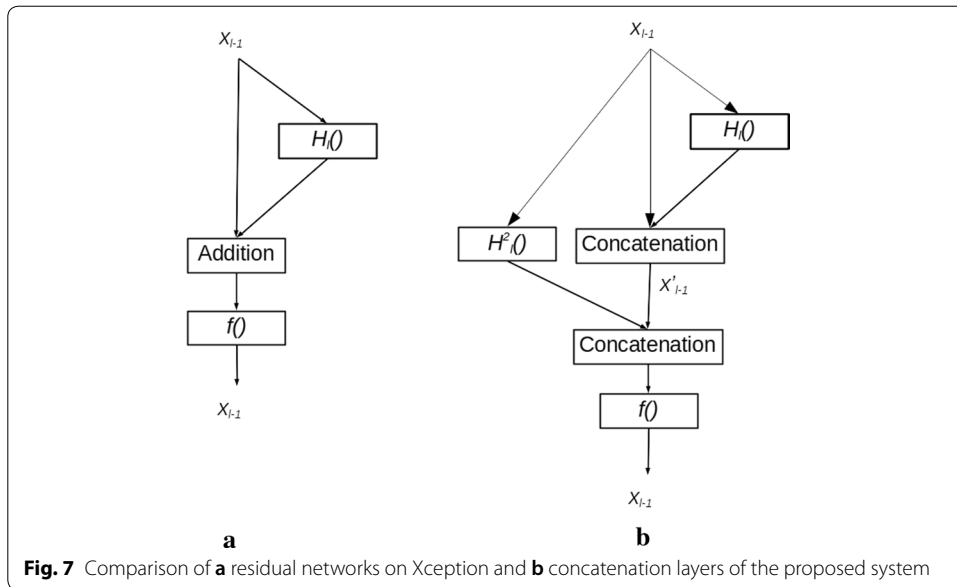


Table 1 Architectures of the proposed model

Blocks	Sub	Types	Output size	# Params
Initialization	–	Conv2D 3 × 3	31 × 31 × 64	1728
	–	Conv2D 3 × 3	29 × 29 × 64	36,864
Compact block 1	Branch 1	$\left[\begin{array}{l} \text{Prev. layer} \\ \text{Conv2D } 3 \times 3 \end{array} \right]$	15 × 15 × 192	368,896
	Branch 2	Conv2D 1 × 1	15 × 15 × 320	8192
Transition	–	Conv2D 3 × 3	8 × 8 × 128	369,920
Compact block 2	Branch 1	$\left[\begin{array}{l} \text{Prev. layer} \\ \text{Conv2D } 3 \times 3 \end{array} \right]$	4 × 4 × 384	885,248
	Branch 2	Conv2D 1 × 1	4 × 4 × 640	32,768
Transition	–	Conv2D 3 × 3	2 × 2 × 256	1,477,120
Compact block 3	Branch 1	$\left[\begin{array}{l} \text{Prev. layer} \\ \text{Conv2D } 3 \times 3 \end{array} \right]$	2 × 2 × 768	3,539,968
	Branch 2	Conv2D 1 × 1	1 × 1 × 1280	131,072
Transition	–	Conv2D 3 × 3	1 × 1 × 256	2,954,240
Classification layer	–	Glob. Av. pool.	256 × 1	–
	–	11 FC layer	11 × 1	2827

The detail implementations of the proposed architectures is shown in Fig. 8 and Table 1. Our networks is comprise of two convolutional layers, then followed by three compact modules, with transition modules in between, and then Global average pooling followed by output layers. The transition modules are built from a convolutional layer, batch normalization, and average pooling. We refer this architectures as ComNet.

Experimental setup

Dataset

We used subset of Database published in [22]. The database itself contains 54,306 images of plant leaves with total of 38 class labels. There are 14 plants in the dataset. From the data, we selected three plants: Apple, Corn, and Potato due to our limited computation resources. There were a total of 9,176 data for the experiments. The original size of the dataset is 256 × 256 pixels. We rescaled them into 64 × 64. The sample images of the dataset is shown in Fig. 10.

We also develop a dataset for tea diseases. We collected 11,367 images of tea leaves which comprise of four classes: one healthy class and 6 types of diseases that are commonly find in tea. They are tea plants with blister blight, leafhopper attacks, looper caterpillars attacks, mosquito bugs attacks, and yellow-mite attacks. The data are collected at Research Center for Tea and Cinchona, Gambung, West Java, Indonesia. The data are collected using two digital cameras and five smartphone cameras. All images are taken indoor with only room lighting. The data are collected during various hours from 8 a.m. in the morning until 5.30 p.m. in the afternoon. The data are scaled into 256 × 256 and then rescaled down into 64 × 64 for the experiments. This dataset in an extension of dataset that we published in [28, 60]. The sample images of this dataset is shown in Fig. 9.

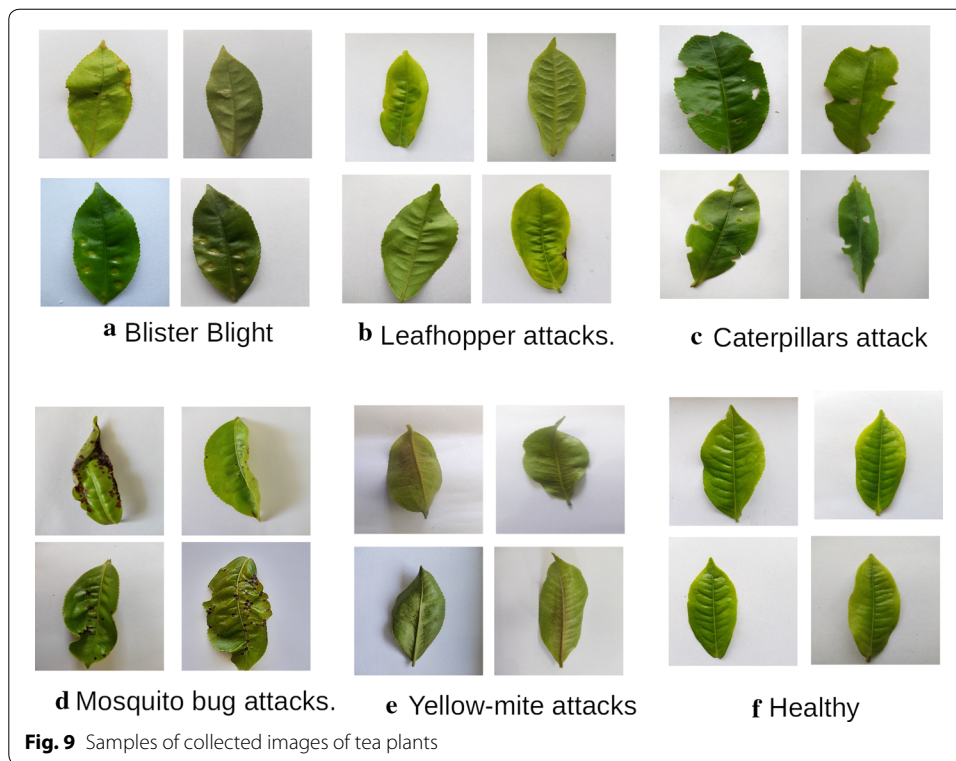


Table 2 List of plants and class labels used in the experiments with the number of data for each class

Plants	Class	Causes	# Data
Apple	Apple cedar rust	<i>Gymnosporangium juniperi-virginiae</i>	276
	Apple scab	<i>Venturia inaequalis</i>	630
	Apple black rot	<i>Botryosphaeria obtusa</i>	621
	Healthy	–	1645
Corn	Corn gray leaf spot	<i>Cercospora zea-maydis</i>	513
	Corn common rust	<i>Puccinia sorghi</i>	1192
	Corn northern leaf blight	<i>Exserohilum turcicum</i>	985
	Healthy	–	1162
Potato	Potato early blight	<i>Alternaria solani</i>	1000
	Potato late blight	<i>Phytophthora infestans</i>	1000
	Healthy	–	152
Tea	Blister blight	<i>Exobasidium vexans</i>	984
	Leafhopper attacks	<i>Empoasca</i> sp.	3224
	Caterpillar attacks	<i>Looper caterpillars</i>	1541
	Mosquito bug attacks	<i>Helopeltis</i> spp.	2140
	Yellow-mite attacks	<i>Polyphagotarsonemus latus</i>	2151
	Healthy	–	1327

The distribution of the data for each plant and class label is shown in Table 2. We apply various image transformations to the test data: Gaussian blur with 5×5 kernel size, median blur with median size is set to 5, 90 degrees rotations, 180 degrees rotation, scaled-down to 32×32 and 48×48 . The sample images for training are shown in

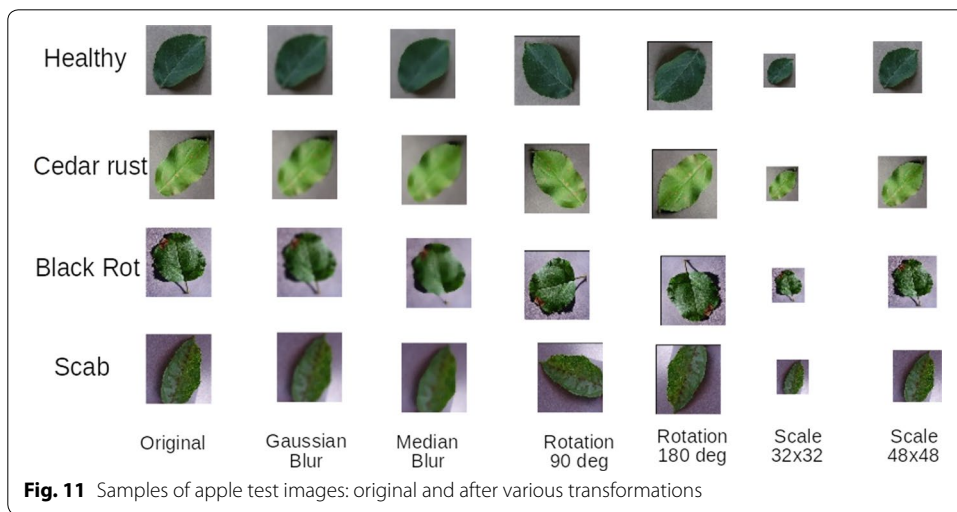
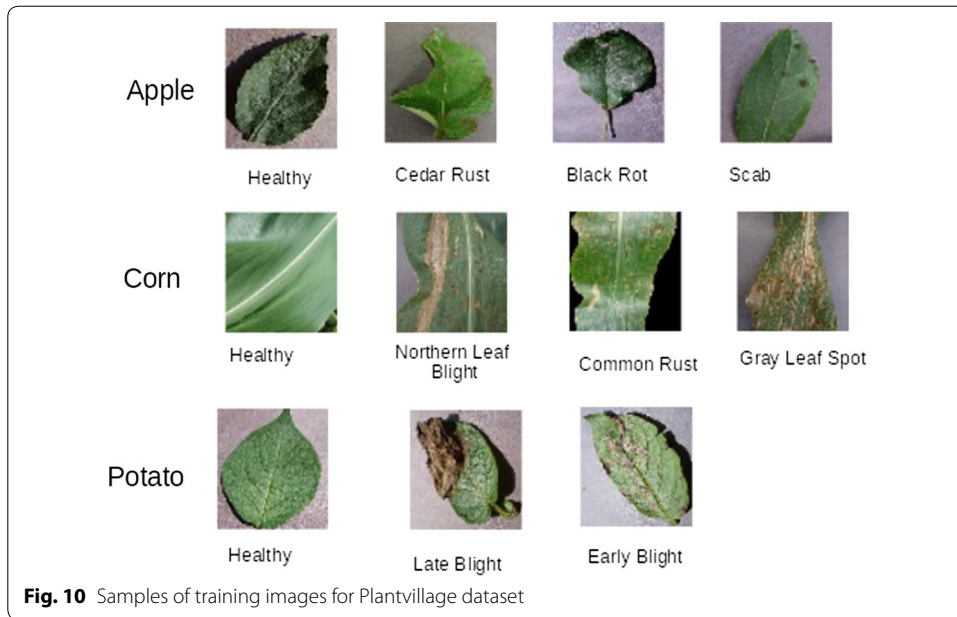


Fig. 10 while the sample of the test data (apple) with the resulting transformed images are depicted in Fig. 11.

Experimental configurations

For the experiments, we use 10-fold cross validations on each model. For learning rate, we use 0.00001 and Adam method[27] for adaptive learning and cross-entropy as loss function. For Plantvillage, we trained all six architectures for all three plants altogether and set the epochs into 100. We also train all architectures for tea dataset with the same configurations as above.

Table 3 The number of parameters (in millions) and the average training time (in seconds) of an epoch for each CNN architectures

Architectures	Num. parameters	Training time	
		Plantvillage	Tea
AlexNet	29.78	2.59	3.86
VGGNet	39.93	7.00	10.35
MobileNet	3.24	3.51	5.43
ResNet	23.59	6.85	10.20
Xception	20.88	7.79	11.78
ComNet	9.66	3.33	4.94

Table 4 The average accuracy of DCNN architectures

Architectures	Ave. accuracy	
	Plantvillage	Tea
AlexNet	94.44	79.82
VGGNet	93.79	81.31
MobileNet	61.17	37.45
ResNet	93.79	70.37
Xception	92.62	73.22
ComNet	96.61	86.17

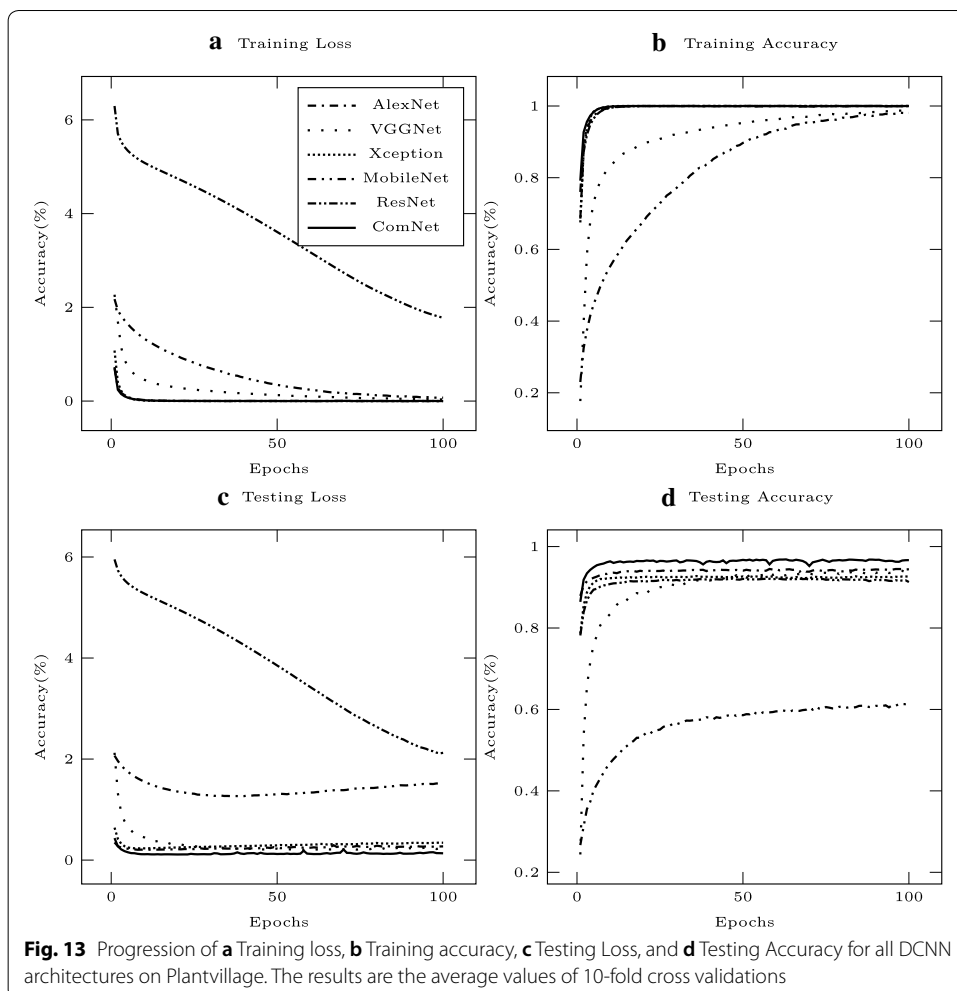
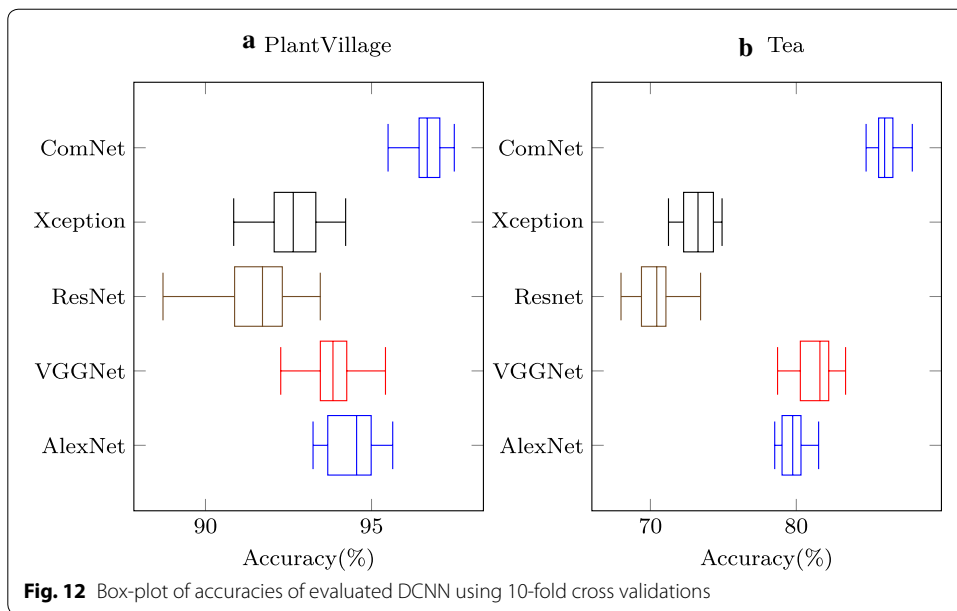
The best performance for each dataset is printed in italics

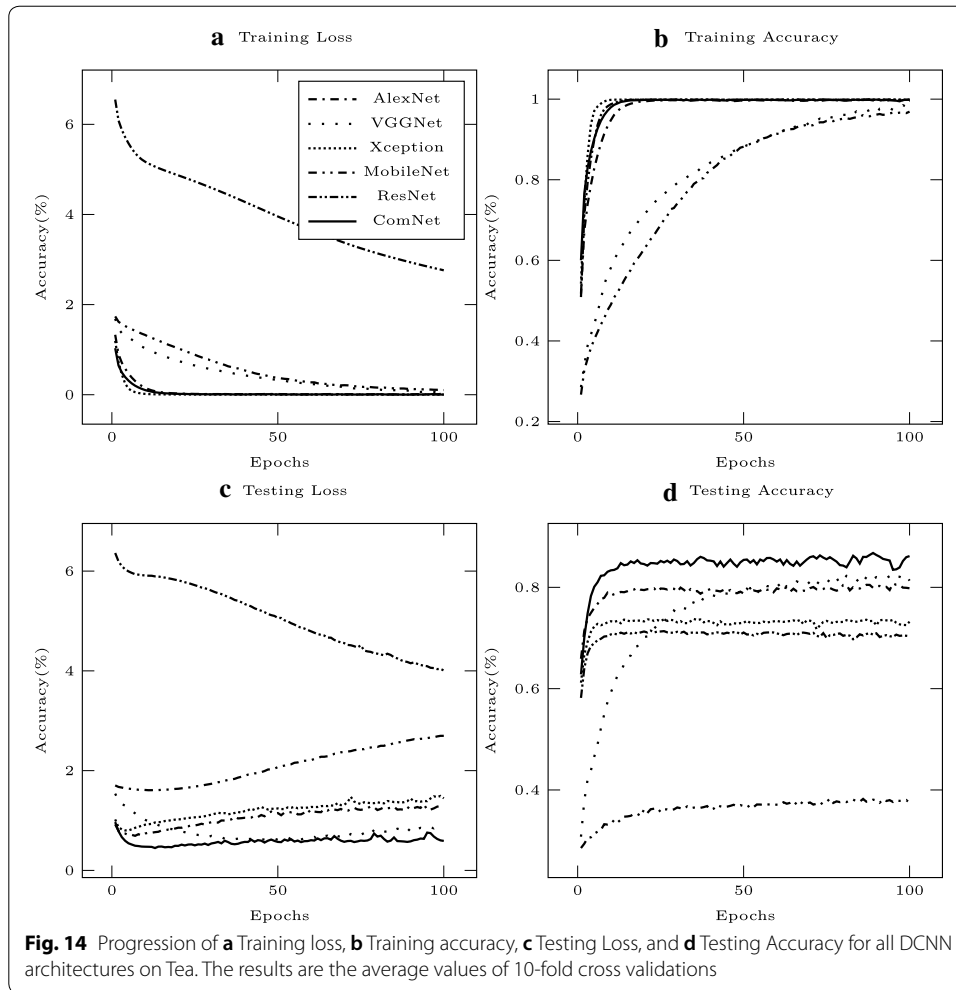
Results and discussions

Number of parameters and training time

As shown in Table 3, of all evaluated architectures, VGGNet has the largest parameters whereas MobileNet is the smallest. This is due to the use of separable convolutional layers. ComNet has only around 9.666 millions parameters. It is much smaller than VGGNet, ResNet, Xception, and AlexNet.

It is also clear that the training times of our method are smaller than VGGNet, ResNet, Xception, and MobileNet, indicating its smaller computational complexity. The computational times depicted in Table 3 are the average computational time over 100 times iterations for both datasets. The experiments are conducted on Intel Xeon E5 2.10GHz CPU and TESLA P100 GPU with 4GB RAM. This is not surprising since our architecture has much smaller number parameters than VGGNet and Xceptions. Interestingly, we notice that compare to MobileNet that has smaller number of parameters, ComNet requires less computational time. This may due to the use of depthwise separable convolutions which are yet to be supported by cuDNN library [40]. Meanwhile, the training time of our method is slightly worse than AlexNet despite having less number of parameters. The use of concatenation layers requires the network to consume more memory space. Since our computational resources have very limited memory space, more communications to the storage are expected, thus making it slower.





Performance comparisons

The results are summarized in Table 4. The results are the average accuracy over 10-fold cross validations. The results clearly show that the proposed method has higher accuracy on both datasets despite its less number of parameters. The boxplot results are shown in Fig. 12. ComNet achieves more than 2% improvements over AlexNet (which has the second best performance) on Plantvillage and almost 5% over VGGNet (the second best) on Tea. Over 10 times repeat, ComNet has quite small range for the boxplot, which suggest the performance of the method is consistent. The results are consistent on both dataset.

The progressions of accuracy and loss of training and testing data for the networks are plotted in Figs. 13 and 14. For Plantvillage, we observe the followings. First, on training data, ResNet has the slowest loss to converge and then followed by MobileNet, and VGGNet. ComNet and Xception are comparable in terms of how fast the loss to converge. On the progression of the accuracy, MobileNet and VGGNet are the slowest to converge. This is as expected as their losses are also slow to converge. Interestingly, the accuracy of training data for ResNet is pretty quick to converge despite the slow losses progression. Meanwhile on testing data, MobileNet appears that it fails to learn and as the results, the performance is the worst among

all the networks. It may be stuck in local minimum for the solutions and larger learning rate may be needed for it to work.

For tea dataset, we notice similar results as Plantvillage for training data. Meanwhile for testing data, we notice that the losses are slowly increasing when the epochs are larger than around 25 for MobileNet, Xception, and AlexNet. This indicates that the networks may already be overfit. As the consequences, the accuracies on testing data are also slowly decreasing. MobileNet also fails to learn for this case.

Our results confirm very promising results that DCNN could be used to recognize plant diseases even when trained with low resolution data. We achieve the accuracy more than 90 % for Plantvillage except for MobileNet. Lower accuracies are obtained for Tea dataset. The results strongly indicate the need for more data for tea. This is due to high variations of data acquisition for tea. The results also show that the DCNN architectures could be used without any usage of complex feature engineering.

Evaluations on the robustness

Table 5 shows the performance of evaluated DCNN when tested with transformed images: blurred, rotated, or scaled down. We transformed the image data using Gaussian Blur with 5×5 kernel (notated as GauBlr), Median Blur with size of 5 (notated as MedBlr), 90 degrees rotation (notated as Rot90), 180° rotation (notated as Rot180), scaled down to 32×32 (notated as Sc32), and scaled down to 48×48 (notated as Sc48). In most experiments, the transformations cause drop in performance. We notice blurring and rotations are found to cause huge drops in performance and scaled-down to have the least effect. When we test on images that are scaled down to 48×48 , the performances are only slightly worse in most cases. This should be expected as we can see a convolutional layers act as a scaling-down operations on image. DCNN aims to learn the abstraction of image data with various sizes to capture various resolution of the data. So a more robust to scaling down operations are expected.

Table 5 The robustness of evaluated DCNN architectures against images transformations

Dataset	Architecture	Conditions					
		GauBlr	MedBlr	Rot90	Rot180	Sc32	Sc48
Plantvillage	AlexNet	62.83	65.76	79.73	81.06	86.06	93.49
	VGG	62.46	66.63	77.37	78.83	81.54	91.98
	MobileNet	46.66	48.12	51.98	55.29	57.37	61.18
	ResNet	65.44	66.52	75.72	64.51	89.05	91.12
	Xception	63.16	65.49	75.78	64.77	87.65	92.04
	ComNet	<i>72.49</i>	<i>74.00</i>	<i>84.20</i>	<i>85.71</i>	<i>93.45</i>	<i>95.90</i>
Tea	AlexNet	57.54	63.83	58.38	53.34	74.31	78.22
	VGG	54.68	60.54	62.08	60.57	71.75	78.57
	MobileNet	34.77	36.45	30.50	32.13	38.00	37.84
	ResNet	57.23	62.84	46.33	42.54	67.67	69.88
	Xception	56.72	64.83	48.66	46.79	69.44	71.46
	ComNet	<i>66.56</i>	<i>75.08</i>	<i>62.21</i>	<i>60.09</i>	<i>83.20</i>	<i>85.40</i>

The best performance for each transformation is printed in italics

We observed that ComNet is consistently more robust than other networks. The two stages of concatenation process and average pooling may contribute to these performances. Mostly, the diseases are identified by the spots found in the leaves. At higher layers with adding and global average pooling operations where larger convolution windows are used, these operations may produce outputs similar to a blurred version of the image, contributing to its robustness. Need to be noted however, our architectures have not been evaluated on realistic “noisy” data as the data are only transformed artificially. Evaluations on more realistic scenarios are needed.

Conclusions

We propose a DCNN architecture for plant diseases detection in this paper using two branches of skip connections. We compared it with other 5 popular DCNN architectures: AlexNet, ResNet, Xception, MobileNet, and VGGNet. We found that our method is consistently better than the reference methods even with smaller number of parameters and faster training time. It is also more robust when it is tested with blurred, rotated, and scaled-down image. However, we should say that the methods have not been evaluated with data with very different conditions as the actual conditions on the fields. Therefore, adding different types of data collected from different conditions are substantial to improve the robustness of the systems on various environmental conditions. This is in our future plans.

It is worth noting that the plant diseases are not meant to replace the actual diagnosis from experts instead it meant to supplement that. Since machine learning methods could only predict with some uncertainty, laboratory test remain the most reliable way to diagnose the plant diseases. However, the implementation could be used to be a help for smallholder farmers who may find it difficult to have fast response from experts

Abbreviations

DCNN: Deep convolutional neural networks; DL: Deep learning; CIP: International Potato Center; CNN: Convolutional neural networks; ComNet: Compact Networks; FFN: Feed-forward network; GLCM: Gray level co-occurrence matrix; GWT: Gabor wavelet transform; HOG: Histogram of Oriented Gradients; ILSVRC: ImageNet Large Scale Visual Recognition Competition; k-NN: k-Nearest neighbor; LBP: Linear Binary Pattern; LVQ: Learning Vector Quantization; PCA: Principle components analysis; SIFT: Scale-Invariant Feature Transform; SURF: Speeded Up Robust Features; SVM: Support Vector Machines.

Acknowledgements

The experiments are conducted in High Performance Computing Facilities of Research Center for Informatics, Indonesian Institute of Sciences.

Authors' contributions

HFP made substantial contribution in formulating the concept, design, carried-out experimentation, analysis and interpretation of the data; ES, VZ, AR participated in carrying-out the experiments and statistical analysis of the results; RBSK, AH, RSY, DK, and AS are participated in coding; FF and VPR validated the dataset. All authors are contributed to the data collections. All authors read and approved the final manuscript.

Funding

This work is partially funded by INSINAS grant from the Indonesian Ministry of Research, Technology, and Higher Education.

Availability of data and materials

The Plantvillage: <https://github.com/spMohanty/PlantVillage-Dataset>. The tea dataset that are used during the current study are not publicly available due to it is in the process of agreement between Research Center for Informatics and Research Institute for Tea and Cinchona but are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests

Author details

¹ Reserch Center for Informatics, Indonesian Institute of Sciences, Bandung, Indonesia. ² Reserch Center for Electronics and Telecommunication, Indonesian Institute of Sciences, Bandung, Indonesia. ³ Research Institute for Tea and Cinchona, Indonesian Agency for Agricultural Research and Development, Gambung, Indonesia.

Received: 5 April 2020 Accepted: 20 July 2020

Published online: 05 August 2020

References

1. Ali H, Lali M, Nawaz MZ, Sharif M, Saleem B. Symptom based automated detection of citrus diseases using color histogram and textural descriptors. *Comput Electron Agric*. 2017;138:92–104.
2. Badnakhe MR, Deshmukh PR. An application of k-means clustering and artificial intelligence in pattern recognition for crop diseases. In: *International conference on advancements in information technology*. 2011.
3. Barker J, Vincent E, Ma N, Christensen H, Green P. The pascal chime speech separation and recognition challenge. *Comput Speech Lang*. 2013;27(3):621–33.
4. Bay H, Ess A, Tuytelaars T, Van Gool L. Speeded-up robust features (surf). *Comput Vis Image Understand*. 2008;110(3):346–59.
5. Belasque J Jr, Gasparoto M, Marcassa L. Detection of mechanical and disease stresses in citrus plants by fluorescence spectroscopy. *Appl Optics*. 2008;47(11):1922–6.
6. Chaerle L, Van Der Straeten D. Imaging techniques and the early detection of plant stress. *Trends Plant Sci*. 2000;5(11):495–501.
7. Chollet F. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint*. 2017;1610–02357.
8. Cirecsan DC, Giusti A, Gambardella LM, Schmidhuber J. Mitosis detection in breast cancer histology images with deep neural networks. In: *International conference on medical image computing and computer-assisted intervention*. Berlin: Springer; 2013. p. 411–8.
9. Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: *Proceedings of the 25th international conference on Machine learning*. New York: ACM; 2008. p. 160–7.
10. Cortes C, Vapnik V. Support-vector networks. *Mach Learn*. 1995;20(3):273–97.
11. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: *international conference on computer vision & pattern recognition (CVPR'05)*, vol. 1. IEEE Computer Society; 2005. p. 886–93.
12. Deng L, Li J, Huang JT, Yao K, Yu D, Seide F, Seltzer ML, Zweig G, He X, Williams JD, et al. Recent advances in deep learning for speech research at microsoft. In: *ICASSP*, vol. 26. Berlin: Springer; 2013. p. 64.
13. Feng J, Liang My, Zhao B, et al. Multispectral imaging system for the plant diseases and insect pests diagnosis. *Spectrosc Spect Anal*. 2009;29(4):1008–12.
14. Feng J, Zhao B, et al. Cucumber diseases diagnosis using multispectral imaging technique. *Spectrosc Spect Anal*. 2009;29(2):467–70.
15. Ferentinos KP. Deep learning models for plant disease detection and diagnosis. *Comput Electron Agric*. 2018;145:311–8.
16. Fuentes A, Yoon S, Kim S, Park D. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*. 2017;17(9):2022.
17. Fukushima K, Miyake S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In: *Competition and cooperation in neural nets*. Berlin: Springer; 1982. p. 267–85.
18. 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*. 2016. p. 770–8.
19. Ho TK. Random decision forests. In: *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE; 1995. p. 278–82.
20. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. 2017. *arXiv preprint* [arXiv:1704.04861](https://arxiv.org/abs/1704.04861).
21. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017. p. 4700–8.
22. Hughes D, Salathé M, et al. An open access repository of images on plant health to enable the development of mobile disease diagnostics. 2015. *arXiv preprint* [arXiv:1511.08060](https://arxiv.org/abs/1511.08060).
23. Iandola F, Moskewicz M, Karayev S, Girshick R, Darrell T, Keutzer K. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint* [arXiv:1404.1869](https://arxiv.org/abs/1404.1869) (2014).
24. International Potato Center: International potato center (cip): Annual report 1996 (1996).
25. Kaiser L, Gomez AN, Chollet F. Depthwise separable convolutions for neural machine translation. *arXiv preprint* [arXiv:1706.03059](https://arxiv.org/abs/1706.03059) (2017).
26. Khaled AY, Abd Aziz S, Bejo SK, Nawi NM, Seman IA, Onwude DI. Early detection of diseases in plant tissue using spectroscopy-applications and limitations. *Appl Spectrosc Rev*. 2018;53(1):36–64.
27. Kingma DP, Ba J. Adam: A method for stochastic optimization. *arXiv preprint* [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014).
28. Krisnandi D, Pardede HF, Yuwana RS, Zilvan V, Heryana A, Fauziah F, Rahadi VP. Diseases classification for tea plant using concatenated convolution neural network. *Commun Inf Technol J*. 2019;13(2).
29. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. 2012. p. 1097–105.
30. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436.
31. LeCun Y, Haffner P, Bottou L, Bengio Y. Object recognition with gradient-based learning. In: *Shape, contour and grouping in computer vision*. Berlin: Springer; 1999. p. 319–45.

32. Liew O, Chong P, Li B, Asundi A. Signature optical cues: emerging technologies for monitoring plant health. *Sensors*. 2008;8(5):3205–39.
33. Lowe DG. Distinctive image features from scale-invariant keypoints. *Int J Comput Vis*. 2004;60(2):91–110.
34. MacKenzie DJ, McLean MA, Mukerji S, Green M. Improved RNA extraction from woody plants for the detection of viral pathogens by reverse transcription-polymerase chain reaction. *Plant Dis*. 1997;81(2):222–6.
35. Martinelli F, Scalenghe R, Davino S, Panno S, Scuderi G, Ruisi P, Villa P, Stroppiana D, Boschetti M, Goulart LR, et al. Advanced methods of plant disease detection. A review. *Agron Sustain Dev*. 2015;35(1):1–25.
36. Meroni M, Rossini M, Picchi V, Panigada C, Cogliati S, Nali C, Colombo R. Assessing steady-state fluorescence and PRI from hyperspectral proximal sensing as early indicators of plant stress: The case of ozone exposure. *Sensors*. 2008;8(3):1740–54.
37. Mohanty SP, Hughes DP, Salathé M. Using deep learning for image-based plant disease detection. *Front Plant Sci*. 2016;7:1419.
38. Ojala T, Pietikainen M, Harwood D. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In: *Proceedings of 12th international conference on pattern recognition*, vol. 1. IEEE; 1994. p. 582–5.
39. Oppenheim D, Shani G, Erlich O, Tsrur L. Using deep learning for image-based potato tuber disease detection. *Phytopathology* pp. PHYTO–08. 2019.
40. Orsic M, Kreso I, Bevandic P, Segvic S. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019. p. 12607–16.
41. Owomugisha G, Melchert F, Mwebaze E, Quinn JA, Biehli M. Machine learning for diagnosis of disease in plants using spectral data. In: *Proceedings on the international conference on artificial intelligence (ICAI). The Steering Committee of the world congress in computer science, Computer . . .*; 2018. p. 9–15.
42. Prasad S, Peddoju SK, Ghosh D. Multi-resolution mobile vision system for plant leaf disease diagnosis. *Signal Image Video Process*. 2016;10(2):379–88.
43. Ramcharan A, McCloskey P, Baranowski K, Mbilinyi N, Mrisho L, Ndalawha M, Legg J, Hughes DP. A mobile-based deep learning model for cassava disease diagnosis. *Front Plant Sci*. 2019;10:272.
44. Rumpf T, Mahlein AK, Steiner U, Oerke EC, Dehne HW, Plümer L. Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. *Comput Electron Agricult*. 2010;74(1):91–9.
45. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, et al. Imagenet large scale visual recognition challenge. *Int J Comput Vis*. 2015;115(3):211–52.
46. Sainath TN, Mohamed AR, Kingsbury B, Ramabhadran B. Deep convolutional neural networks for Ivcsr. In: *2013 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. New York: IEEE; 2013. p. 8614–8.
47. Sankaran S, Mishra A, Ehsani R, Davis C. A review of advanced techniques for detecting plant diseases. *Comput Electron Agric*. 2010;72(1):1–13.
48. Sarikaya R, Hinton GE, Deoras A. Application of deep belief networks for natural language understanding. *IEEE/ACM Trans Audio Speech Lang Process*. 2014;22(4):778–84.
49. Savary S, Ficke A, Aubertot JN, Hollier C. Crop losses due to diseases and their implications for global food production losses and food security. 2012.
50. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 2014.
51. Strange RN, Scott PR. Plant disease: a threat to global food security. *Annu Rev Phytopathol*. 2005;43.
52. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, inception-resnet and the impact of residual connections on learning. In: *Thirty-first AAAI conference on artificial intelligence*. 2017.
53. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015. p. 1–9.
54. Tai AP, Martin MV, Heald CL. Threat to future global food security from climate change and ozone air pollution. *Nat Clim Change*. 2014;4(9):817.
55. Veit A, Wilber MJ, Belongie S. Residual networks behave like ensembles of relatively shallow networks. In: *Advances in neural information processing systems*. 2016. p. 550–558.
56. Voulodimos A, Doulamis N, Doulamis A, Protopapadakis E. Deep learning for computer vision: a brief review. *Computational intelligence and neuroscience*. 2018;2018.
57. Wetterich CB, Kumar R, Sankaran S, Belasque Junior J, Ehsani R, Marcassa LG. A comparative study on application of computer vision and fluorescence imaging spectroscopy for detection of Huanglongbing citrus disease in the USA and Brazil. *J Spectrosc*. 2012;2013.
58. Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017. p. 1492–500.
59. Yang Y, Chai R, He Y. Early detection of rice blast (pyricularia) at seedling stage in nipponbare rice variety using near-infrared hyper-spectral image. *Afr J Biotechnol*. 2012;11(26):6809–17.
60. Yuwana RS, Suryawati E, Zilvan V, Ramdan A, Pardede HF, Fauziah F. Multi-condition training on deep convolutional neural networks for robust plant diseases detection. In: *2019 international conference on computer, control, informatics and its applications (IC3INA)*. 2019. p. 30–35. <https://doi.org/10.1109/IC3INA48034.2019.8949580>.
61. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: *European conference on computer vision*. Berlin: Springer; 2014. p. 818–33.
62. Zhang M, Qin Z, Liu X, Ustin SL. Detection of stress in tomatoes induced by late blight disease in California, USA, using hyperspectral remote sensing. *Int J Appl Earth Observ Geoinf*. 2003;4(4):295–310.

Publisher's Note

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