Computers and Electronics in Agriculture 165 (2019) 104948

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Depthwise separable convolution architectures for plant disease classification

Kamal KC, Zhendong Yin*, Mingyang Wu, Zhilu Wu

School of Electronics and Information Engineering, Harbin Institute of Technology, China

ARTICLE INFO

Keywords:
Deep learning
Separable convolution
MobileNet
Plant disease classification
Reduced parameters

ABSTRACT

Convolutional neural network has a huge partake and is still a dominating tool in the field of computer vision. In this study, we introduce a model with depthwise separable convolution architecture for plant disease detection based on images of leaves. We present two versions of depthwise separable convolution comprising two varieties of building blocks. Training and testing of the models were performed on a subset of publicly available PlantVillage dataset of 82,161 images containing 55 distinct classes of healthy and diseased plants. These depthwise separable convolutions achieved less accuracy and high gain in convergence speed. Several models were trained and tested, of which Reduced MobileNet achieved a classification accuracy of 98.34% with 29 times fewer parameters compared to VGG and 6 times lesser than that of MobileNet. However, MobileNet outperformed existing models with 36.03% accuracy when testing the model on a set of images taken under conditions different from those of the images used for training. Thin models were also introduced, which showed effective trade-off between latency and accuracy. The satisfactory accuracy and small size of this model makes it suitable for real-time crop diagnosis in resource constrained mobile devices.

1. Introduction

Crop diseases are a major cause of famine and food insecurity around the world. Early diagnosis and prevention of such diseases are the best solutions to increase crop yield. An essential aspect of disease monitoring is early and accurate identification. The key responsibility is not only to find the abnormality in plants but also its type. Visual examination is considered an efficient method for early disease detection; however, it is prone to human error due to tiresome continuous monitoring. Feature extraction and pattern recognition in machine learning help to spot the disease type and severity of infestation of the disease in plants. Automated quality analysis of plant health via images of plant leaves, characterized by color, shape, and size is an accurate and reliable method for increased productivity.

Identifying objects of any genre based on their visual traits is a tedious task. One instance of such a task is the plant identification of LifeCLEF2017 (Goeau et al., 2017). A classical machine learning approach involves the extraction of a discriminative set of visual features followed by a machine learning algorithm to associate correct labels to given attributes. The effectiveness of the method is highly dependent on the preciseness of feature design and the learning algorithm. Since the rise of deep learning, manual intervention of feature extraction has been outdated. A survey (Kamilaris and Prenafeta-Boldú, 2018) indicates that deep learning offers better performance than popular image processing techniques. Deep learning has a deep impact in various fields, one among them being the agro-industry. It helps to determine crop quality and increase crop quantity in real time without manual intervention. Deep learning algorithms yield high accuracy at the expense of data abundance. In the initial days of deep learning, researchers aimed to create deep networks by the addition of neural network layers for increased accuracy. However, the introduction of different architectures of networks has been quite promising. Different techniques to develop slim and accurate deep neural networks has become crucial for real-world applications, especially for those employed in embedded systems (Dong et al., 2017). Recent researches (Han et al., 2016; Narang et al., 2017), which emphasize on pruning deep networks at the cost of marginal loss in accuracy while achieving a sizable reduction in model size, infer that deep networks could be severely overparameterized and simple reduction of a number of hidden units while maintaining the models' dense connection structure can be a viable solution. Researchers (Zhu and Gupta, 2017) found that for a deep convolutional neural netwrok (CNN), large sparse models consistently outperformed small dense models and achieved up to 10x reduction in the number of non-zero parameters with minimal loss in accuracy. Because both theoretical analysis and empirical experiments have shown the evidence of redundancy in several deep models, it is possible

* Corresponding author at: School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China.
E-mail address: yinzhendong@hit.edu.cn (Z. Yin).

https://doi.org/10.1016/j.compag.2019.104948
Received 15 December 2018; Received in revised form 28 July 2019; Accepted 11 August 2019
0168-1699/ © 2019 Published by Elsevier B.V.

to compress deep neural networks with or without loss in prediction by pruning parameters with carefully designed criteria (Han et al., 2016).

In contrast to tedious and prolonged hand-tailored feature extraction in classical machine learning algorithms, deep learning algorithms provide automated feature extraction. CNNs have shown consistent and superior results compared to state-of-art solutions employing hand-crafted features (Lee et al., 2015). Classical approaches in agriculture started with plant leaf detection (Agarwal et al., 2006) based on leaf shape, leaf texture, vein shape, and lamina based methods. Many systems have been proposed for plant identification based on leaf images (Park et al., 2008; Wang et al., 2008; Wu et al., 2006; Liu et al., 2016; Yanikoglu et al., 2014). Deep learning approaches (Mehdipour-Ghazi et al., 2017; Reyes et al., 2015) achieved high accuracy while classifying plants based on leaf images. Plant identification by ResNet26 on the BJFU100 dataset demonstrated that deep learning is a promising technology for smart forestry (Sun et al., 2017).

Numerous feature extraction processes in adjunction with several machine learning algorithms were used to classify plant diseases based on leaf images. Spectral features were extracted by applying visible-near infrared spectroscopy in the field of articulation with quadratic discriminant analysis classification algorithm to detect Huanglongbing in citrus orchards (Sankaran et al., 2011). Binary classification was performed using Support Vector Machine with a kernel-based function on 200 RGB images to classify two types of diseases on tomato plants with 91.5% accuracy (Mokhtar et al., 2015). A robust model to identify, classify, and quantify a diverse set of foliar stresses in soybean, using a large and diverse dataset of unseen test samples (around 600 per class), achieved an overall classification accuracy of 94.13% (Ghosal et al., 2018).

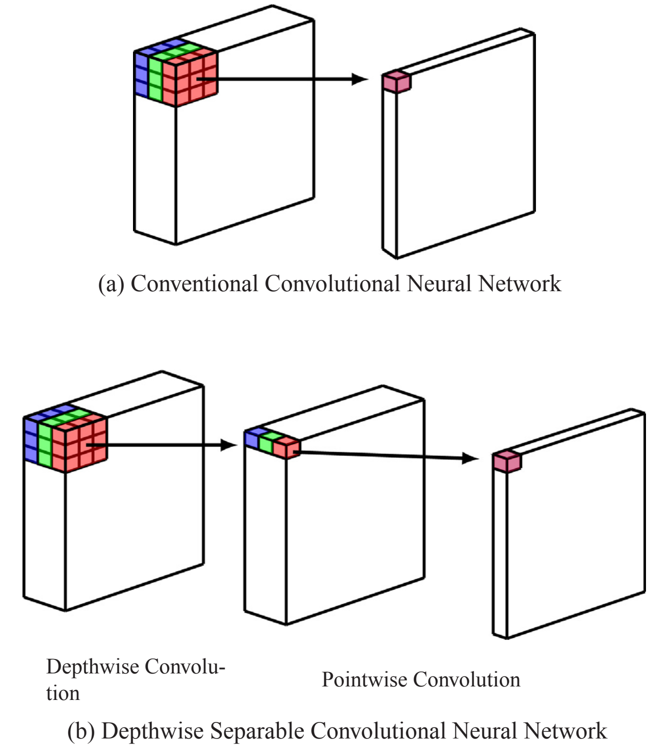
Deep learning in agro-industry for plant disease classification is a recent trend. Because of the popularity of CNN in image classification, it has been ubiquitous in the detection of diseases in plant species. A detailed study on the application of various architectures, which were used in ImageNet large-scale visual recognition challenge (ILSVRC) (Russakovsky et al., 2015), was used to classify 26 classes of diseased plants from the PlantVillage dataset, which achieved an accuracy of 99.35% (Mohanty et al., 2016) while using transfer learned GoogleNet. Plant disease classification of 87,484 plant leaf images from 58 distinct classes of [plant, disease] combinations were classified using a CNN-based deep learning algorithm with 99.53% accuracy (Ferentinos, 2018). Models built from scratch and those using transfer learning were compared. CNN is considered as a black box whose features learned for classification are unknown. To unravel the black box, deconvolution network was deployed to visualize the chosen features. CNN was trained on MalayaKew Leaf Dataset and a classification accuracy of 99.5% was achieved. The authors concluded that primary and secondary venation structures were the key features extracted during classification (Lee et al., 2015). To ensure automated low cost early detection of crop diseases, deep convolution models with high accuracy, satisfactory inference time, and model size suitable for real-time crop state diagnosis on a large scale with limited hardware capabilities were deployed (Bhatt et al., 2017).

2. Materials and methods

2.1. Model architecture

Our proposed system is based on a depthwise separable convolutional network, which has been explicitly incorporated in MobileNet (Howard et al., 2017). Depthwise separable convolutions are factorable convolutions comprising of depthwise convolution, which convolves kernels of each filter with each input channel, and pointwise convolution, which mixes the resulting output channels.

2.1.1. Depthwise separable convolution


(a) Conventional Convolutional Neural Network

Depthwise Convolution

Pointwise Convolution

(b) Depthwise Separable Convolutional Neural Network

Fig. 1. Convolution mechanism in Conventional CNN and Depthwise Separable CNN. Depthwise Separable convolution splits standard convolutional into two distinct steps. Depthwise convolution performs convolution within a single depth slice, as seen by three different colours, where each colour represents the depth. Pointwise Convolution merges the information across the whole depth.

For the $l-th$ layer in a network with 3D input tensor x^l such that $x^l \in \mathrm{IR}^{H^l \times W^l \times D^l}$, where H^l, W^l, and D^l are the height, width, and depth of the input for layer l, a triplex index set (i^l, j^l, d^l) is specified to point to any specific element in x^l. The triplet (i^l, j^l, d^l) refers to one element in x^l, which is in the $d-th$ channel and at spatial location (i^l, j^l).

Assuming D filters of spatial size span $H \times W$ are used from filter bank f such that $f \in \mathrm{IR}^{H \times W \times D^l \times D}$, where D^l is a receptive field in x^l, the convolution output y is given as

$$y_{i^{l+1},j^{l+1},d} = \sum_{i=0}^{H}\sum_{j=0}^{W}\sum_{d=0}^{D} f_{i,j,d} \times x^{l}_{i^{l+1}+i,j^{l+1}+j,d}. \tag{1}$$

In Eq. (1), $x^{l}_{i^{l+1}+i,j^{l+1}+j,d}$ refers to the element of x^l indexed by triplets $(i^{l+1}+i, j^{l+1}+j, d)$.

The basic idea behind depthwise separable convolution is to split feature learning, operated by standard convolutions, over two simpler steps: a spatial feature learning step and a channel combination step, as shown in Fig. 1.

$$y_{i^{l+1},j^{l+1},d} = \sum_{d=0}^{D} f_d \times \sum_{i=0}^{H}\sum_{j=0}^{W} f_{i,j} \times x^{l}_{i^{l+1}+i,j^{l+1}+j} \tag{2}$$

In Eq. (2), f_d is a 1×1 convolution filter.

2.1.2. Core building blocks

A deep CNN model is built on convolutional layers. Various state-of-the-art deep learning models such as LeNet (Lecun et al., 1998), AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan and Zisserman, 2014), Inception (Szegedy et al., 2015), ResNet (He et al., 2016), Xception (Chollet, 2017), etc., which performed well in ILSVRC (Russakovsky et al., 2015), are built on a similar concept. However, it is difficult to deploy such models on mobile platforms due to large computational latency. To overcome this hurdle, depthwise separable convolutions are introduced, which increase the efficiency in trade-off

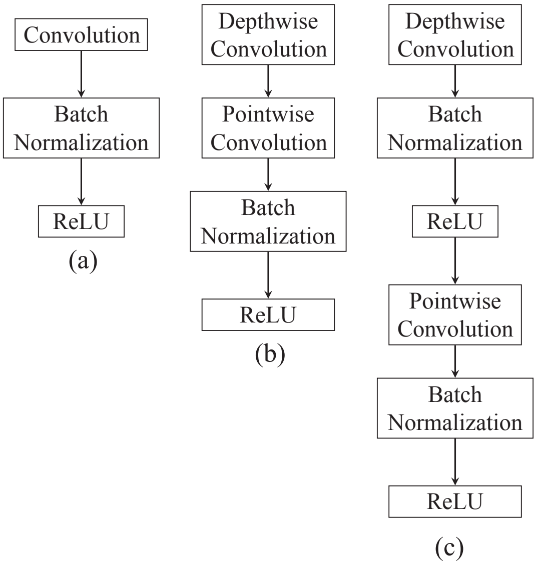


Fig. 2. Building blocks of deep neural network. (a) Standard Convolution as core layer, (b) Depthwise Separable Convolution as core layer, (c) Depthwise Separable Convolution as proposed in MobileNet as core layer.

between latency and accuracy.

The proposed depthwise separable convolution comprises of two varieties of building blocks. Fig. 2(c) shows the building block used in MobileNet architecture, in which batch normalization and a non-linear activation operation, rectified linear unit (ReLU), are introduced after each depthwise and pointwise convolution. Fig. 2(c) shows a building block similar to that of MobileNet (Howard et al., 2017), which is quantization-friendly (Sheng et al., 2018) and removes batch normalization and ReLU after depthwise convolution.

2.2. Methodology

We created two models from scratch using depthwise separable convolution, namely Modified MobileNet and Reduced MobileNet, instead of conventional convolution layers. We compared the results with MobileNet, which we created from scratch. The PlantVillage dataset was split into three sets: training, validation, and testing. Validation data was used to tune the network parameters and hyperparameters to prevent overfitting. Unseen test data was used to get a generalized measure of classification accuracy. The train-validation-test data split percentage of 70–20-10 was used to avoid overfitting (Mohanty et al., 2016). Different types of optimizers, viz., SGD, Adam, and Nadam, were adopted to evaluate training and testing accuracies. Table 1 shows the training hyperparameters used while employing various optimizers during training.

2.2.1. Model design

We experimented on three models: Modified MobileNet, Reduced MobileNet, and MobileNet. AlexNet and VGG had an input size of 224 × 224, two of the former models had a 150 × 150 input size to reduce the computational cost, and MobileNet had an input size of 224 × 224. AlexNet and VGG architectures, defined in Tables 2 and 3 respectively, were deployed to train MNIST and CIFAR-10 datasets. All the convolutional layers were followed by the ReLU activation function. Modified MobileNet comprises of a building block as shown in Fig. 2(b), whereas Reduced MobileNet makes use of a building block as shown in Fig. 2c. Tables 4 and 5 define the model architectures of Modified MobileNet and Reduced MobileNet, respectively, used in the study.

Table 1
CNN training hyperparameters for various optimizers.

Hyperparameter	SGD	Adam	Nadam
Batch Size	20	20	32
Momentum	0.9	–	–
Decay	None	$1e^{-06}$	0.004
Learning Rate	0.001	0.001	0.002
$\beta 1$	–	0.9	0.9
$\beta 2$	–	0.999	0.999
ϵ	–	$1e^{-08}$	$1e^{-08}$

Table 2
AlexNet architecture.

Layer Size/Stride	Output Shape	Parameters
5,5 Conv, 96/S1	223,223,96	7296
3,3 Maxpooling/S2	111,111,96	0
3,3 Conv, 256/S1	111,111,256	221440
3,3 Maxpooling/S2	55,55,256	0
Dropout(0.5)		
2∗(3,3 Conv, 384/S1)	55,55,384	2212608
3,3 Conv, 256/S1	55,55,256	884992
3,3 Maxpooling/S2	27,27,256	0
Dropout(0.5)		
Global Average Pooling	1,1,256	0
Dense	1,1,16	448
Softmax	1,1,16	0

Table 3
VGG architecture.

Layer Size/Stride	Output Shape	Parameters
2∗(3,3 Conv, 64/S1) 3,3 Maxpooling/S2 Dropout(0.5)	37,37,64	38720
2∗(3,3 Conv, 128/S1) 3,3 Maxpooling/S2 Dropout(0.5)	18,18,128	221440
3∗(3,3 Conv, 256/S1) 3,3 Maxpooling/S2 Dropout(0.5)	9,9,256	1475328
3∗(3,3 Conv, 512/S1) 3,3 Maxpooling/S2 Dropout(0.5)	4,4,512	5899776
3∗(3,3 Conv, 512/S1 3,3 Maxpooling/S2) Dropout(0.5)	2,2,512	7079424
Global Average Pooling	1,1,512	0
Dense	1,1,16	48
Softmax	1,1,16	0

2.2.2. Thinner models

At times, a specific application might require a small model (Howard et al., 2017). We introduce a parameter α called width

Table 4
Modified MobileNet architecture.

Layer Size/Stride	Output Shape	Parameters
3,3 Conv,32/S2	74,74,32	896
3,3 Maxpooling/S2	36,36,32	0
2∗(3,3 SeparableConv, 128/S1	32,32,128	21920
3,3 Maxpooling/S2)	15,15,128	0
2∗(3,3 SeparableConv, 256/S1	11,11,256	101760
3,3 Maxpooling/S2)	5,5,256	0
2∗(3,3 SeparableConv, 512/S1	1,1,512	400128
Global Average Pooling	1,1,512	0
Dense	1,1,16	32
Softmax	1,1,16	0

Table 5
Reduced MobileNet architecture.

Layer Size/Stride	Output Shape	Parameters
3,3 Conv,32/S2	74,74,32	992
3,3 DepthwiseConv, 32/S1	75,75,64	2720
1,1 PointwiseConv, 64/S1		
3,3 DepthwiseConv, 64/S2	38,38,128	9536
1,1 PointwiseConv, 128/S1		
3,3 DepthwiseConv, 128/S1	38,38,128	18560
1,1 PointwiseConv, 128/S1		
3,3 DepthwiseConv, 128/S2	19,19,256	35456
1,1 PointwiseConv, 256/S1		
3,3 DepthwiseConv, 256/S1	19,19,256	69888
1,1 PointwiseConv, 256/S1		
3,3 DepthwiseConv, 256/S2	10,10,512	136448
1,1 PointwiseConv, 512/S1		
3,3 DepthwiseConv, 512/S1	10,10,512	270848
1,1 PointwiseConv, 512/S1		
Global Average Pooling	1,1,512	0
Dense	1,1,16	8208
Softmax	1,1,16	0

multiplier to construct small and computationally inexpensive models. Width multiplier condenses a network uniformly at each layer. The computational cost of a standard convolutional layer with input feature map of size $D_F \times D_F \times M$ and kernel of size $D_K \times D_K \times M \times N$, where D_F and D_K are the spatial dimensions of a square input feature map and kernel, respectively, and M and N are the number of input and output channels, respectively, is

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F. \tag{3}$$

The computational cost of the depthwise separable convolutional layer with a depthwise convolutional kernel size of $D_K \times D_K \times M$ and a 1×1 pointwise convolution is

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F \tag{4}$$

which reduces the computational cost by 8 or 9 times while using a 3×3 depthwise separable convolution (Howard et al., 2017). After implementing depth multiplier α, the computational cost becomes

$$D_K \cdot D_K \cdot \alpha M \cdot D_F \cdot D_F + \alpha M \cdot \alpha N \cdot D_F \cdot D_F, \tag{5}$$

which reduces the cost and number of parameters quadratically by roughly α^2.

2.2.3. *Dataset*

The dataset used in our experiments was obtained from the PlantVillage dataset (Hughes and Salathé, 2015). It contains 82,161 plant leaf images of varied sizes from 24 plants divided into 55 classes. The dataset contains images with a clean background and cluttered background, as shown in Fig. 3. Clean background images consist of isolated leaves with uniform background, whereas cluttered background images comprise part or full images of plants taken in a natural background. The number of images per class varies from 43 to 6359. This dataset was further split into three different sets. PlantLeaf1 contains 18 classes that comprise of images with cluttered background. None of the images in this dataset contain laboratory-conditioned images. PlantLeaf2 contains 11 classes, which constitute both clean and cluttered images. Clean background images were used for training, while cluttered background images were used for testing in this dataset. PlantLeaf3 contains 16 classes from 11 plants. These classes contain both clean and cluttered images, while the number of images per class ranged from 892 to 5507. This dataset contains 10 classes from 10 different crop species and 6 classes from tomato plants infected with different diseases. The number of classes and images for each PlantLeaf dataset is detailed in Table 6. Discrepancy in data within a class is determined by the difference between between training and testing data. Detailed information about all the PlantLeaf datasets is provided in Supplementary Information 1.

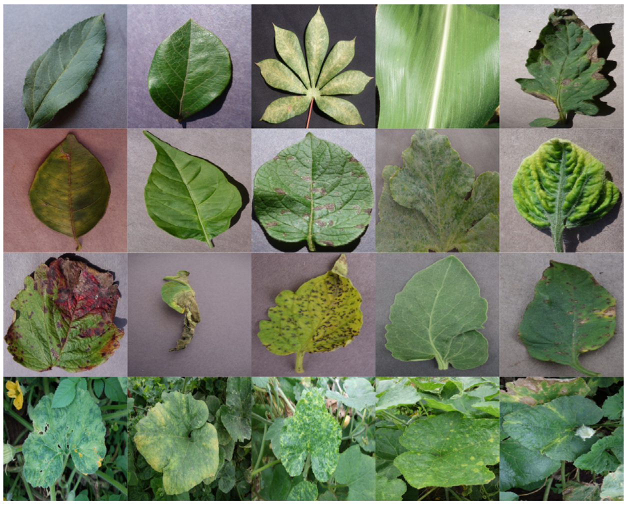


Fig. 3. Some samples from our dataset. The dataset includes images with clean as well as cluttered backgrounds (bottom row).

Table 6
Set of datasets.

Dataset	Number of classes	Number of images	Discrepancy
PlantVillage	55	82161	Low
PlantLeaf1	18	18517	Low
PlantLeaf2	11	23110	High
PlantLeaf3	16	32241	Low

To generalize the model and ensure a robust model, these image datasets were augmented using different data augmentation processes, such as flipping, random crops, rotations, shifts, and a combination of these techniques. The aim of data augmentation is to prevent overfitting by training the model to large data created artificially.

To benchmark a model, it is essential to test the model against a standard dataset. Database of such datasets are readily available. We tested our models against the MNIST (Lecun et al., 1998) and CIFAR-10 (Krizhevsky and Hinton, 2009) datasets. MNIST is a database of handwritten digits. It has a training set of 60,000 and a test set of 10,000 examples. The input image is a grayscale image of size 28×28 and the dataset contains 10 classes. The CIFAR-10 dataset consists of 60,000 32×32 RGB images in 10 classes with 6,000 images per class. There are 50,000 training images and 10,000 test images. The classes are completely mutually exclusive.

2.2.4. *Benchmarking with standard datasets*

We first evaluated the effectiveness of various models on the MNIST and CIFAR-10 datasets. MNIST and CIFAR-10 are widely used for benchmarking computer vision algorithms in the field of machine learning. In the experiment, the networks were trained on the training set using Keras (Chollet et al., 2015) for 50 and 200 epochs. We initialized the weights and trained all networks from scratch. We used the SGD optimizer with a mini-batch size of 20 due to resource constraints and a momentum of 0.9. The learning rate was set to 0.001, which decayed by an order of magnitude $1e-6$ every epoch. For data augmentation, we adopted real-time data augmentation, which loops over data in batches. The testing results on these two datasets after training and testing on various models is provided as Supplementary Information (Table S1 and Table S2).

3. Results

Modified MobileNet and Reduced MobileNet achieved an accuracies of 97.65 and 98.34%, respectively, for 55 classes of the PlantVillage

Table 7
Comparing accuracies (%) of PlantVillage Dataset.

Model	Test Accuracy	Number of parameters (in millions)
AlexNet (Ferentinos, 2018)	99.44	3.3
VGG (Ferentinos, 2018)	99.53	14.7
Modified MobileNet	97.65	0.5
Reduced MobileNet	98.34	0.54
MobileNet	98.65	3.2

leaves dataset. MobileNet achieved an accuracy of 98.65% for the same dataset [Table 7]. These models slightly underperformed compared to VGG (Ferentinos, 2018), which achieved an accuracy of 99.53%. However, MobileNet outperformed both GoogleNet using transfer learning (Mohanty et al., 2016), which achieved an accuracy of 31.4%, and VGG (Ferentinos, 2018), which achieved an accuracy of 33.27%, with 36.03% of accuracy on the PlantLeaf2 dataset. Reduced MobileNet achieved an accuracy of 32.42%, which is 2% higher when using Adam instead of SGD as the optimizer in the model on the PlantLeaf2 dataset. We used the Nadam optimizer for MobileNet.

While training and testing real-condition images in the PlantLeaf1 dataset using the Nadam optimizer with hyperparameters from Table 1, accuracy slightly reduced to 97.64% [Table 8], which suggested that similarity in training and testing data is an essential aspect in deep learning, while background noise slightly affects the model's performance. Training MobileNet on the PlantLeaf3 dataset yielded the highest accuracy of 99.62% [Table 9]. Reduced MobileNet outperformed GoogleNet (Mohanty et al., 2016) by attaining an accuracy of 99.37%. Table 10 shows the time taken per epoch for the models to complete training for different datasets. It also depicts the epoch in which the model performed the best. Fig. 4 shows the accuracies obtained on three datasets (MNIST, CIFAR-10, and PlantLeaf3) by employing two different models (Reduced MobileNet and MobileNet) and their variants by introducing width multiplier ($\alpha = \{0.25, 0.5, 0.75, 1\}$). Accuracy increases for higher values of α; however, the number of parameters and hence the computational cost also increase [Fig. 4].

Table 8
Test Accuracies (%) on PlantLeaf1 dataset for various optimizers.

Models	SGD	Adam	Nadam
MobileNet	97.07	97.26	97.64
Reduced MobileNet	96.43	98.04	98.2

Table 9
Test Accuracies (%) on PlantLeaf3 dataset for different width multiplier.

Width Multiplier(α)	Reduced MobileNet	MobileNet
1	99.37	99.62
0.75	98.52	98.76
0.5	98.02	98.82
0.25	97.35	96.17

Table 10
Training Statistics (epoch time in seconds @ best performing epoch) for best performing models.

Model	PlantVillage	PlantLeaf1	PlantLeaf2	PlantLeaf3
Reduced MobileNet	1640@163	160@195	228@138	468@80
Modified MobileNet	1591@165	158@189	227@159	455@73
MobileNet	1846@148	390@178	306@127	597@104

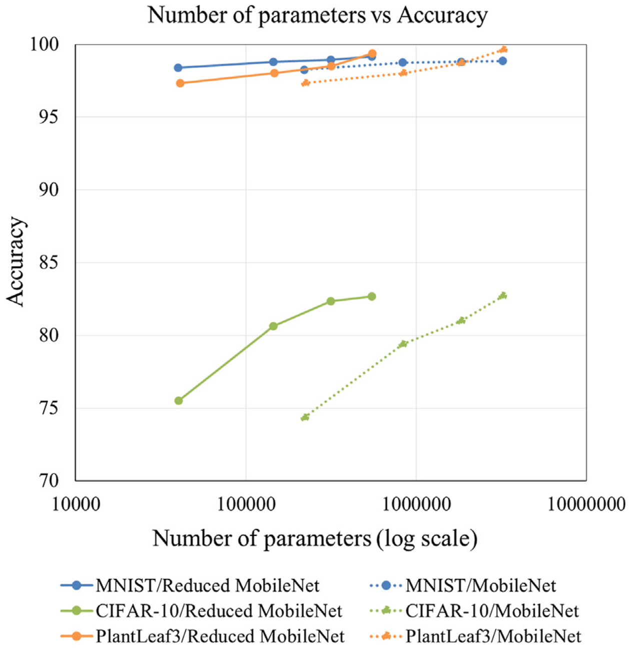


Fig. 4. Test Accuracy vs Number of Parameters. Accuracies are plotted for various $\alpha = \{0.25, 0.5, 0.75, 1\}$ for MobileNet and Reduced MobileNet model. Bold line represent Reduced MobileNet whereas broken lines represent MobileNet. Three different colors for three datasets are used (Blue, Green and Orange represent MNIST, CIFAR-10 and PlantLeaf3 dataset respectively). The points on the graph from left to right for a particular [colour, line style] combination corresponds to α = 0.25, 0.5, 0.75, 1 respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4. Conclusion

In this work, various deep learning models were developed and compared based on different CNN architectures for efficient classification of plant diseases based on simple leaf images of healthy and diseased plants. A separable CNN model matched the accuracy of conventional CNN with drastically lesser parameters, thereby making it an ideal model to be used in embedded devices. The most successful model architecture, MobileNet, achieved a success rate of 98.65% with roughly six times lesser parameters than VGG. In addition, Reduced MobileNet, which is a pruned version of MobileNet, with the final five convolution layers retracted, attained similar accuracy in comparison to MobileNet with substantially reduced parameters. Final recurring layers of the network, which were used for high-level feature extraction, seem to be insignificant at some point and contribute to computational overhead. Moreover, different optimizers yielded different results. Compared to SGD, Adam and Nadam performed better with faster convergence rate. The trade-off between accuracy and computational cost led to the development of small and inexpensive models by introducing width multiplier. Appropriate models can be selected based on resource constraints. The experiment on the PlantLeaf1 dataset demonstrates that coherent data results in a desirable outcome, while noisy data marginally affects the aftermath. The disassociation in data within a class can immensely decrease the accuracy of the model. The results from training and testing models on the PlantLeaf2 dataset reveal that alienation in intra class data causes high loss in accuracy. In addition, these experiments on the PlantLeaf3 dataset illustrate that efficiency is dependent on intra class correlation. The data in PlantLeaf3 had higher intra class correlation compared to inter class correlation.

The classification of 55 different classes from 24 plant species using separable convolution is efficient task and the first of its kind. However, implementation of this model on large databases could be time consuming; hence, outperforming the other models could be a challenging aspect. The proposed deep learning approach showed higher efficacy on the available dataset, and its potential depends on the quality and quantity of available data. This study explored the potential of efficient network architecture and various network models, which can easily satisfy the design requirements for mobile and embedded vision applications.

Funding

The research presented in this article was supported by the National Natural Science Foundation of China (Grant Nos. 61471142, 61601145, 61571167 and 61871157).

Authors Contributions

K.KC designed, performed the research, and wrote the paper. Y.Z. collected the data and helped in editing the paper. M.W. helped with data arrangement. Y.Z. and Z.W. procured the funding.

Declaration of Competing Interest

The authors declare no competing financial interests.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.compag.2019.104948.

References

Agarwal, G., Belhumeur, P., Feiner, S., Jacobs, D., John Kress, W., Ramamoorthi, R., Bourg, N., Dixit, N., Ling, H., Mahajan, D., Russell, R., Shirdhonkar, S., Sunkavalli, K., White, S., 2006. First steps toward an electronic field guide for plants. Taxon 55, 597–610.

Bhatt, P., Sarangi, S., Pappula, S., 2017. Comparison of cnn models for application in crop health assessment with participatory sensing. In: IEEE Global Humanitarian Technology Conference (GHTC), pp. 1–7.

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.

Chollet, F., et al., 2015. Keras. https://keras.io.

Dong, X., Chen, S., Pan, S., 2017. Learning to prune deep neural networks via layer-wise optimal brain surgeon. In: Adv. Neural Informat. Process. Syst. pp. 4857–4867.

Ferentinos, K., 2018. Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agricul. 145, 311–318.

Ghosal, S., Blystone, D., Singh, A.K., Ganapathysubramanian, B., Singh, A., Sarkar, S., 2018. An explainable deep machine vision framework for plant stress phenotyping. Proc. Nat. Acad. Sci. 115, 4613–4618.

Goeau, H., Bonnet, P., Joly, A., 2017. Plant identification based on noisy web data: the amazing performance of deep learning (lifeclef 2017). In: CLEF 2017-Conference and Labs of the Evaluation Forum, pp. 1–13.

Han, S., Mao, H., Dally, W.J., 2016. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In: 4th International Conference on Learning Representations, ICLR 2016. http://arxiv.org/abs/1510.00149.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

Howard, A.G., 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:1704.04861.

Hughes, D.P., Salathé, M., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. arXiv preprint arXiv:1511.08060, 1–13.

Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: A survey. Comput. Electron. Agric. 147, 70–90.

Krizhevsky, A., Hinton, G., 2009. Learning multiple layers of features from tiny images. Technical Report. University of Toronto.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86, 2278–2324. https://doi.org/10.1109/5.726791.

Lee, S.H., Chan, C.S., Wilkin, P., Remagnino, P., 2015. Deep-plant: Plant identification with convolutional neural networks. In: 2015 IEEE International Conference on Image Processing (ICIP), pp. 452–456. https://doi.org/10.1109/ICIP.2015.7350839.

Liu, N., Kan, J., et al., 2016. Plant leaf identification based on the multi-feature fusion and deep belief networks method. J. Beijing Forestry Univ. 38, 110–119.

Mehdipour-Ghazi, M., Yanikoglu, B.A., Aptoula, E., 2017. Plant identification using deep neural networks via optimization of transfer learning parameters. Neurocomputing 235, 228–235.

Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7, 1419.

Mokhtar, U., Ali, M.A.S., Hassanien, A.E., Hefny, H., 2015. Identifying two of tomatoes leaf viruses using support vector machine. In: Information Systems Design and Intelligent Applications. Springer India, New Delhi, pp. 771–782.

Narang, S., Elsen, E., Diamos, G., Sengupta, S., 2017. Exploring sparsity in recurrent neural networks. arXiv preprint arXiv:1704.05119.

Park, J., Hwang, E., Nam, Y., 2008. Utilizing venation features for efficient leaf image retrieval. J. Syst. Softw. 81, 71–82.

Reyes, A.K., Caicedo, J.C., Camargo, J.E., 2015. Fine-tuning deep convolutional networks for plant recognition, in: CLEF (Working Notes).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., 2015. Imagenet large scale visual recognition challenge. Int. J. Comput. Vision 115, 211–252.

Sankaran, S., Mishra, A., Maja, J.M., Ehsani, R., 2011. Visible-near infrared spectroscopy for detection of huanglongbing in citrus orchards. Comput. Electron. Agric. 77, 127–134.

Sheng, T., Feng, C., Zhuo, S., Zhang, X., Shen, L., Aleksic, M., 2018. A quantization-friendly separable convolution for mobilenets. In: 2018 1st Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2). IEEE, pp. 14–18.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Sun, Y., Liu, Y., Wang, G., Zhang, H., 2017. Deep learning for plant identification in natural environment. Computat. Intell. Neurosci. 2017, 1–6.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., 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.

Wang, X.F., Huang, D.S., Du, X., Xu, H., Heutte, L., 2008. Classification of plant leaf images with complicated background. Appl. Math. Comput. 205, 916–926.

Wu, Q., Zhou, C., Wang, C., 2006. Feature extraction and automatic recognition of plant leaf using artificial neural network. Adv. Artif. Intell. 3, 5–12.

Yanikoglu, B., Aptoula, E., Tirkaz, C., 2014. Automatic plant identification from photographs. Machine Vision Appl. 25, 1369–1383.

Zhu, M., Gupta, S., 2017. To prune, or not to prune: exploring the efficacy of pruning for model compression. arXiv preprint arXiv:1710.01878.