Computers and Electronics in Agriculture xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

A comparative study of fine-tuning deep learning models for plant disease identification

Edna Chebet Too[a,*], Li Yujian[a], Sam Njuki[a], Liu Yingchun[b]

[a] Beijing University of Technology, Department of Computer Science, Beijing, China
[b] Academy of Forest Inventory and Planning, State Forestry Administration, Beijing 100714, China

ARTICLE INFO

Keywords:
Plant disease identification
Deep learning
Convolutional neural networks
Image recognition
Fine-tuning

ABSTRACT

Deep learning has recently attracted a lot of attention with the aim to develop a quick, automatic and accurate system for image identification and classification. In this work, the focus was on fine-tuning and evaluation of state-of-the-art deep convolutional neural network for image-based plant disease classification. An empirical comparison of the deep learning architecture is done. The architectures evaluated include VGG 16, Inception V4, ResNet with 50, 101 and 152 layers and DenseNets with 121 layers. The data used for the experiment is 38 different classes including diseased and healthy images of leafs of 14 plants from plantVillage. Fast and accurate models for plant disease identification are desired so that accurate measures can be applied early. Thus, alleviating the problem of food security. In our experiment, DenseNets has tendency's to consistently improve in accuracy with growing number of epochs, with no signs of overfitting and performance deterioration. Moreover, DenseNets requires a considerably less number of parameters and reasonable computing time to achieve state-of-the-art performances. It achieves a testing accuracy score of 99.75% to beat the rest of the architectures. Keras with Theano backend was used to perform the training of the architectures.

1. Introduction

Deep learning is currently a remarkably active research area in machine learning and artificial intelligence and has been extensively and successfully applied in numerous fields. Essentially, it is a class of machine learning techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification (Deng and Yu, 2014). Moreover, it has been applied in speech and Audio Processing, Natural Language Processing (NLP) as well as Computer Vision (Deng and Yu, 2014; He et al., 2016; Huang et al., 2016). Additionally, it has been widely applied in many sectors of the world such as Business, agriculture, automotive industry etc. in object detection and image classification (Mohanty et al., 2016; Sladojevic et al., 2016; Dyrmann et al., 2016; Reyes et al., 2015).

There have been breakthroughs for image classification through the deep Convolution Neural Network (CNN). Recently, a number of modifications of CNN Architecture have been proposed with a gradual increase in the number of layers. Some of the architectures include: AlexNet (Krizhevsky et al., 2012), GoogLeNet Inception V3 (Szegedy et al., 2015), Inception V4 (Szegedy et al., 2016), VGG net (Simonyan and Zisserman, 2015), Microsoft ResNet (He et al., 2016), DenseNets (Huang et al., 2016). These deep networks may have difficulties and challenges such as exploding/vanishing gradients and degradation in the training process. Most deeper networks suffer from the degradation problem, where there is a reduction of accuracy when the depth of the network exceeds maximum. Another challenge is the internal covariate shift which is the change of the distribution of the input data to a layer during training. However, a number of optimization techniques have been proposed to deal with the difficulties and challenges satisfactorily, including skip connections (He et al., 2016), transfer learning (Pan and Fellow, 2009), initialization strategies (Mishkin and Matas, 2016), Optimization strategies (Le et al., 2011), batch Normalization (Szegedy and Com, 2015) and layer-wise training (Yu et al., 2016).

Advancement in image classification presents an opportunity to extend the research and application of image processing to the field of agriculture (Mohanty et al., 2016; Sladojevic et al., 2016; Reyes et al., 2015). Deep learning models can now be used in the detection and classification of plant disease using images. A number of different deep learning approaches are currently used for the task of detecting plant disease (Mohanty et al., 2016; Sladojevic et al., 2016). Food security is a major concern with the expected world population growth of more than 9.7 billion by 2050 (Melorose et al., 2015). Plant disease are a threat to food security, therefore, accurate methods are needed to

* Corresponding author.
E-mail address: tooedna@emails.bjut.edu.cn (E.C. Too).

https://doi.org/10.1016/j.compag.2018.03.032
Received 30 October 2017; Received in revised form 28 February 2018; Accepted 27 March 2018

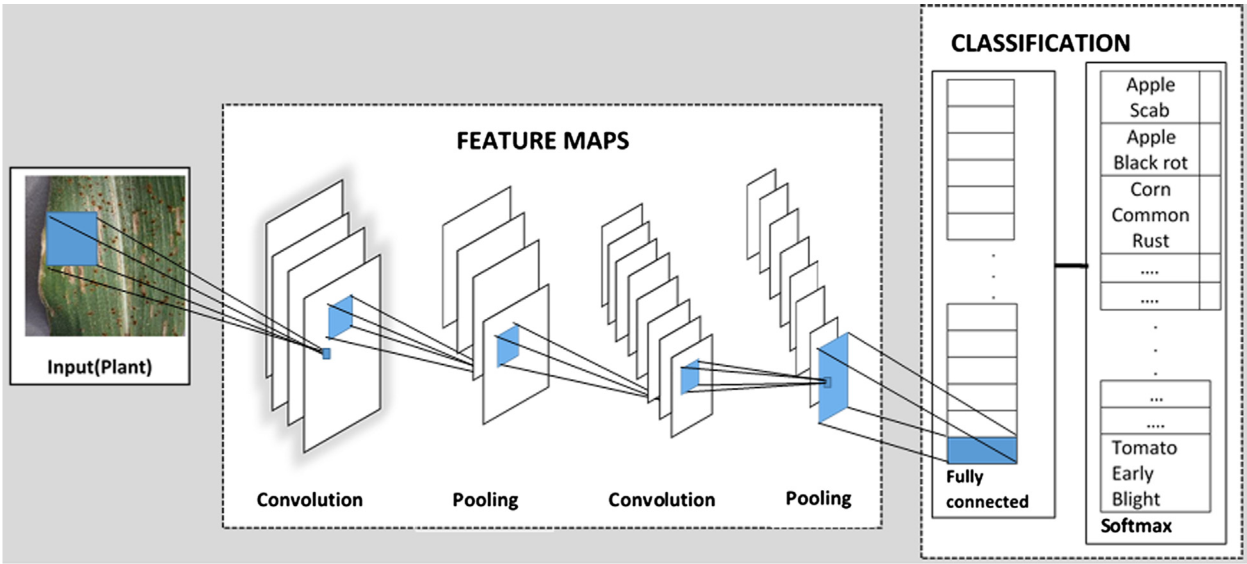

Fig. 1. A typical Convolution Neural Network (CNN) architecture.

identify the diseases so that appropriate measures can be done.

In this study, an empirical analysis of the state-of-the-art deep learning models in the task of detecting and classifying plant disease is done.

The rest of the paper is organized as follows. Section 1.1 looks at related work done in the field of agriculture. In Section 2 we describe some of the existing state-of-the-art deep Convolutional methods as well as the other materials and methodology required to accomplish this task, Section 3 presents the experimental setup as well as the results, Section 4, discussion and conclusion.

1.1. Related work

Several approaches are used in the agricultural field including investigation of plant disease and pests. Deep learning has likewise been applied as well as image processing techniques. Traditional machine learning approaches have been extensively adopted in the agricultural field.

Mohanty et al. (2016) in their work applied Deep learning method to develop a smartphone-assisted disease diagnosis system. They used CNN to training their model using datasets of 54,306 images of healthy and infected plant leaves. CNN was trained to identify 14 crop species and 26 diseases using images. They evaluated the appropriateness of CNN for the classification problem of plants/crop and diseases. They employed two architectures AlexNet [8] and GoogLeNet (Szegedy et al., 2015). Their model achieved an accuracy of 99.35%. Although their model generated state-of-the-art result it performed poorly when it was tested on sets of images taken under different condition.

Similarly, Sladojevic et al. (2016) adopted Deep CNN to the development of plant disease recognition model using leaf images. Their model was able to recognize 14 different types of plant disease from healthy leaves. Additionally, it was able to distinguish plants from their surroundings. They achieved an average of 96.3% accuracy on their experimental analysis.

Equally, deep learning architectures have been used for plants species classification by Dyrmann et al. (2016). In their work, they present a method that can recognize weeds and plant species using colored images. They applied CNN in their work, which was tested on a total 10,413 images with 22 weeds and crop species. CNN model was able to achieve a classification accuracy of 86.2%. The network had a problem classifying some plant species and this is believed to have been due to a low number of training samples for those species.

Another model, called DeepFruits, used in agriculture for fruit detection was proposed by Sa et al. (2016). In their work they present a CNN approach for fruit detection using imagery data. Their goal was to build an accurate, fast and reliable fruit detection system, which is a vital element of agricultural for yield estimation and automated harvesting. They adopted faster R-CNN (Ren et al., 2015) model and referred it to multi-modal Faster R-CNN. They trained their model and was able to achieve an improvement from the previous work of 0.838 precision and recall in the detection of sweet pepper. They retrained to perform the detection of seven fruits, with the entire process taking four hours to annotate and train the new model per fruit (Sa et al., 2016).

Machine learning techniques have equally been applied in plant disease classification. Athanikar and Badar (2016) applied Neural Network to categorize the potato leaf image as either healthy or diseased. Their results showed that BPNN could effectively detect the disease spots and classify the particular disease type with an accuracy of 92%.

Additionally, Wang et al. (2012) did an experimental research to find out a method to realize image recognition of plant disease. Four kinds of neural networks were used to distinguish wheat stripe rust from wheat leaf rust and to distinguish grape downy mildew from grape powdery mildew based on color features, shape features and texture features extracted from the disease images. The results showed that identification and diagnosis of the plant disease could be effectively achieved using Neural networks based on image processing.

Moreover, Samanta et al. (2012) propose image processing methodology to detect scab disease of potato. The images are collected from different potato field and are processed for enhancement. The image segmentation is carried out to get target regions (disease spots). Finally, an analysis of the target regions (disease spots) based on histogram approach to finding the phase of the disease (Samanta et al., 2012).

2. Materials and methods

Deep learning is currently a very active research field in computer vision and image classification. A typical Deep CNN consists of an input and an output or classification layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and in some cases Softmax layer. Most CNN architectures follows the design pattern of LeCun's et al., Lenet-5 architecture (LeCun et al., 1998). A typical CNN architecture that's is applied by a number of architectures is shown in Fig. 1.

Thereafter, a number of architectures have been designed. In this work, an evaluation of the state-of-the-art convolutional neural network and fine-tuning it for the task of plant disease identification and classification using images from PlantVillage is done (Hughes and Salathe, 2015). PlantsVillage contains Openly and freely dataset with 54,306 images, with 26 diseases for 14 crop plants.

The architectures evaluated include VGG 16, Inception V4, ResNet with 50, 101 and 152 layers and DenseNets with 121 layers. Fast and accurate models for plant disease identification are desired so that accurate measures can be applied early.

2.1. Dataset

Deep learning models were evaluated and trained on images of plant leaves with the aim of classifying and identifying disease on images that the model has not seen before. Openly and freely dataset from PlantVillage (Hughes and Salathe, 2015) were used for this study. PlantVillage have 54,306 images, with 26 diseases for 14 crop plants. The images are originally colored images of varied sizes. The images are first resized to 224×224 for VGG net, ResNet and DenseNets architectures. On the other hand, for the Inception V4 architecture the images are resized to 299×299 pixels. Normalization of data is done by dividing all pixel values by 255 to make them compatible with the network's initial values. Furthermore, one hot encoding of target variable or categorical variable is done in order to be used in the models studied.

The data is first split into two. First is the training data and then test data with percentage ratio of 80% and 20% respectively. The choice of the split ratio is based on Mohanty et al. (2016) work. The test set is used for prediction and evaluation of the models.

The training data is further split into two; training and validation data with the ration of 80% and 20% respectively to determine if the model is overfitting. The training set was 34,727 samples, validation set was 8702 samples and testing set of 10,876 samples.

2.2. State-of-the-art deep learning image classifiers

2.2.1. VGG net model

VGG net is CNN model devised by Simonyan and Zisserman (2015) for the ILSVRC-2014 challenge. The model attained a 7.5% top-5 error rate on the validation set which is an outcome that secured them a second place in the competition. Typically, the model is symbolized by its modesty as depicted in Simonyan and Zisserman (2015), with only 3×3 convolutional layers stacked on top of each other in increasing depth. Max pooling handles reducing the size of the volume (downsampling). Additionally, two fully-connected layers each with 4096 nodes and a softmax classifier as shown in their work (Simonyan and Zisserman, 2015).

Fine-tuning the VGG 16 was done by truncating the original softmax layer and replace it with our own. The number of our class labels is 38. Furthermore, a pre-trained model with weights from ImageNet was used. Finally, the model evaluated based on cross-entropy loss and accuracy on the test set.

2.2.2. ResNet

He et al. in their paper (He et al., 2015) introduced the ResNet model which was a basis of ILSVRC 2015 and COCO 2015 classification challenge. Their model won the 1st place on with error rate of 3.57% in the ImageNet classification. The inability of multiple non-linear layers to learn identity mappings and degradation problem motivated the deep residual learning framework (ResNet).

ResNet is a network-in-network (NIN) architecture that relies on many stacked residual units. These residual units are the set of building blocks used to construct the network. A collection of residual unit's forms building blocks that leads to the ResNet Architecture (He et al., 2015). The residual units are composed of convolution, pooling, layers. The architecture is similar to the VGG net (Simonyan and Zisserman, 2015) consisting of 3×3 filters but ResNet, is about 8 times deeper than VGG network. This is attributed due to the usage of global average pooling rather than fully-connected layers. A further update of ResNet (He et al., 2016) was done to obtain more accuracy by updating the residual module to use identity mappings.

A ResNet model with 50,101 and 152 layers as in He et al. (2016) and load it with pre-trained weights from ImageNet was created. Finally, a customized softmax layer was created for the task of plants disease identification.

2.2.3. Inception V4

The "Inception" concept was first introduced in the GoogLeNet architecture by Szegedy et al. (2015).

The subsequent manifestations of GoogLeNet architecture have been referred to Inception *vN* with *N* referring to the version number.

In the paper by Szegedy et al. (2015), proposed Inception V3 architecture which proposes updates to the Inception module to similarly raise ImageNet classification accuracy.

Szegedy et al. (2016) further improved the architecture to give rise to Inception V4. This architecture combines the Inception architecture with residual connections. Their aim being to accelerate the training of Inception networks.

The Inception module is made-up of a pooling layer and convolution layers stacked together. The convolutions are of varied sizes of 1×1, 3×3 and 5×5. Another salient feature of the Inception module is the use of bottleneck layer which is a 1×1 convolutions. The bottleneck layer helps in reduction of computation requirements. Additionally, there is pooling layer is used for dimension reduction within the module. To merge layers a concatenation filter is required as shown in Szegedy et al. (2015). Inception v4 replaces the filter concatenation stage of the Inception architecture with residual connections (Szegedy et al., 2016).

Fine-tuning of GoogLeNet Inception V4 by using pre-trained weights from ImageNet was performed. Additionally, truncation and definition of a new model with Average pooling layer (8×8), dropout and softmax on the top layer was performed.

2.2.4. DenseNet

Huang et al. (2016) in their paper introduced a densely connected convolutional network architecture. To ensure maximum information flow between layers in the network, all layers are connected directly with each other in a feed-forward manner.

For each layer, the feature-maps of all preceding layers are used as inputs and its own feature-maps are used as inputs into all subsequent layers. DenseNets alleviates the problem of the vanishing-gradient problem and has substantially reduced number of parameters (Huang et al., 2016).

For this task of plant disease identification, DenseNets model with 121 layers as described in Huang et al., 2016 was created. Additionally, the model was loaded with pre-trained weights from ImageNet. Finally, another fully-connected model with our own customized softmax on the top layer was created.

2.3. Fine-tuning the models

Fine-tuning is a concept of transfer learning. Transfer learning is a machine learning technique, where knowledge gain during training in one type of problem is used to train in other related task or domain (Pan and Fellow, 2009). In deep learning, the first few layers are trained to identify features of the task. During transfer learning, you can remove last few layers of the trained network and retrain with fresh layers for target job. Fine-tuned learning experiments require a bit of learning, but they are still much faster than learning from scratch (Mohanty et al., 2016). Additionally, they are more accurate compared to models trained from scratch.

To accelerate learning, the CNN models were fine-tuned to identify and classify 38 categories of plant disease with pre-trained models on

Table 1
Accuracy and loss of training, validation and testing and its execution time per epoch.

Model and specifications			10 epochs			30 epochs						
Model	Layers	Params	Training accuracy%	Validation accuracy%	Training loss	Training accuracy	Validation accuracy%	Training loss	Validation Loss	Test Accuracy %	Test Loss	TIME ~ secs
Inception V4		41.2 M	99.89	99.38	0.0272	99.74	98.02	0.0102	0.0673	98.08	0.0686	4042
VGG net	16	119.6 M	67.02	76.12	0.7725	83.86	81.92	0.5069	0.5997	81.83	0.6055	1051
ResNet	50	23.6 M	99.95	99.57	0.0188	99.99	99.67	6.238e-04	0.0159	99.59	0.02177	1583
ResNet	101	42.5 M	99.95	99.53	0.0201	99.99	99.66	4.1611e-04	0.0165	99.66	0.02082	2766
ResNet	152	58.5 M	99.99	97.98	0.2815	100	99.68	2.4844e-04	0.0156	99.59	0.0246	4366
DenseNet	121	7.1 M	99.98	99.71	0.0110	100	99.76	5.6427e-04	0.0107	99.75	0.0159	2165

ImageNet dataset. The ImageNet dataset contains about 1.2 million images and 1000 class categories. On the other hand, PlantVillage dataset is 54,306 images and 38 classes. Thus, the PlantVillage dataset is insufficient to train deep networks hence the use of the pre-trained weights from the ImageNet.

Fine-tuning was done on CNN Inception v4, VGG16, ResNet and DenseNets architecture on PlantVillage dataset without data augmentation.

The models were created and loaded with pre-trained weights from ImageNet. Additionally, truncation of top layer was performed by defining a new fully-connected softmax layer on the top layer.

Additionally, fine-tuning the model was done using stochastic gradient descent (SGD) algorithm and an initial learning rate of 0.001.

2.4. Batch Normalization

Batch Normalization is a technique that helps to minimize the problems of Internal Covariate Shift (Szegedy and Com, 2015). When training Deep Neural Networks, the output of one layer is the input of the next layers. During training of the network, the distribution of the input data to the layers vary significantly over time as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization and makes it notoriously hard to train models with saturating nonlinearities.

Batch Normalization helps to minimize the challenges posed by internal Covariate Shift. The input of each layer is normalized by adjusting the mean and variance of the input across one minibatch. Batch Normalization allows the use of much higher learning rates and less worry about initialization, and in some cases eliminates the need for Dropout. Batch normalization potentially helps in two ways: faster learning and higher overall accuracy (Szegedy and Com, 2015). Batch normalization and ReLU activation function are applied in all the experiments.

2.5. Hardware and software

The experiments were performed on dual Graphics Processing Unit (GPU) mode. The specifications of the machine used: the memory of 16 GB, processor clock 33 MHz, Graphics of Tesla K40c, and the operating system used is Ubuntu 16.04 64 bits.

Python: Python is considered a reasonably comfortable for data science. Many choose python due to its popularity and community support for data science. Raising the community is the primary reason for open sourcing. Consequently, Python supports a number of Deep Learning frameworks.

Keras: Keras is a simple to use neural network library built on top of Theano or TensorFlow that allows developers to prototype ideas very quickly (Chetlur et al., 2014). Keras provides most of the building blocks needed to build reasonably sophisticated deep learning models. It also comes with a great documentation and tons of online resources. It works with python. This framework was used along with the set of weights learned on a very large dataset, ImageNet (https://github.com/liuzhuang13/DenseNet.,; https://keras.io/).

CuDNN: CuDNN is a library for CUDA, developed by NVIDIA, which provides highly tuned implementations of primitives for deep neural networks. CuDNN make deep nets run faster and sometimes using less memory (Chetlur et al., 2014). Thus, CuDNN was configured to work with Theano Backend.

CNMeM: CNMeM is a simple library, developed by NVIDIA, which helps deep learning frameworks in managing of CUDA memory. CNMeM is already integrated into Theano.

OpenCV- OpenCV is a free library for both academic and commercial use that supports and Windows, Linux, Mac OS, iOS and Android and C++, C, Python and Java interfaces. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing (http://opencv.org).

3. Results

3.1. Experiments

3.1.1. Experiment setup

The baseline system used in our evaluation is a workstation GPU Tesla K40c. OpenCV, Keras, Theano, CNMeM and CuDNN library are used for software implementations.

3.1.2. Training

For every experiment, accuracy metric and categorical cross-entropy loss (loss) are used for evaluation of the models. The performances are graphically depicted for each model with accuracy and loss. An overall loss score and accuracy based on the test dataset are computed and used to determine the performance of the models. The results are presented in Table 1. Each of the experiment runs for a total of 10 epochs and 30 epochs. Where the epoch is the number of the training iterations. The choice of the 10 and 30 epoch was done to check which model was able to converge with few iterations and which one suffers from the degradation problem.

The hyper-parameters were standardized on all the networks. All the network models are trained using Stochastic Gradient Descent (SGD), SGD runs faster and converges easily (He et al., 2016). Because of GPU memory constraints, we trained the networks with the Batch size of 16. The learning rate was set to 0.001 for all networks. We used the weight Decay 1e-6 and a Nesterov momentum of 0.9. Batch Normalization (https://keras.io/) technique and ReLU activation function (Glorot et al., 2011) are applied in all the experiments. No data augmentation was done for all the networks.

3.1.3. Results of the experiments

In this study, an assessment of the appropriateness of state-of-the-art deep convolutional neural network for the task of plant disease identification using images was done. Our focus was fine-tuning VGG 16, Inception V4, ResNet with 50,101 and 152 layers and DenseNets with 121 layers.

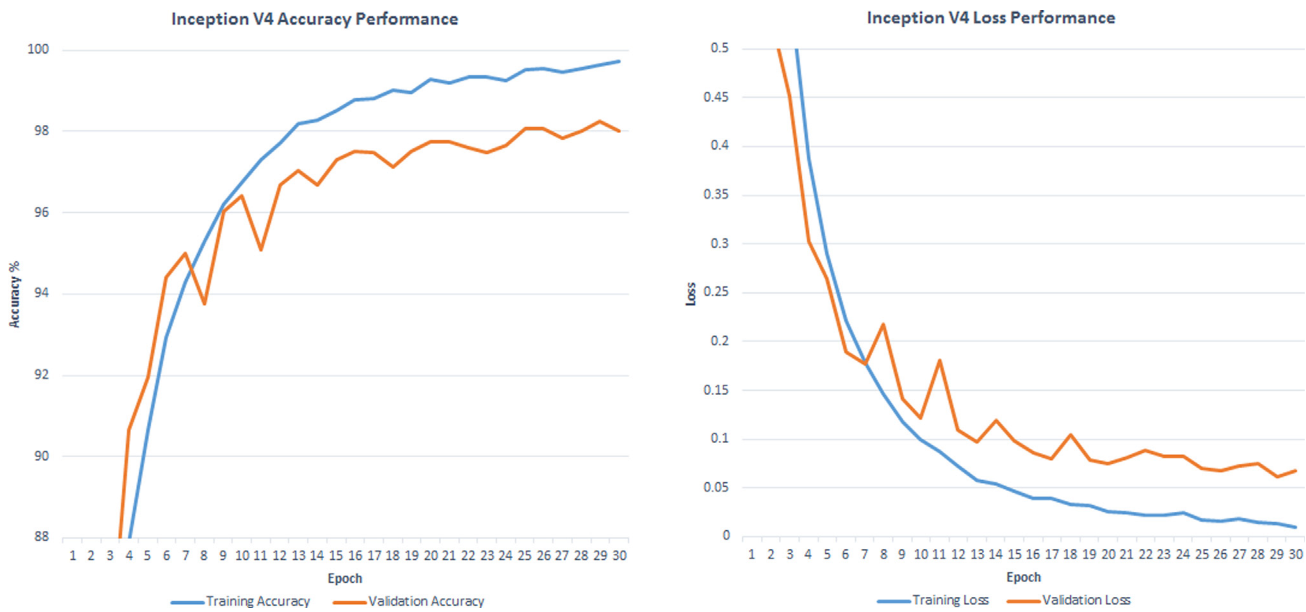

Fig. 2. Inception V4, left is accuracy of the model and right depicts the Model Loss.

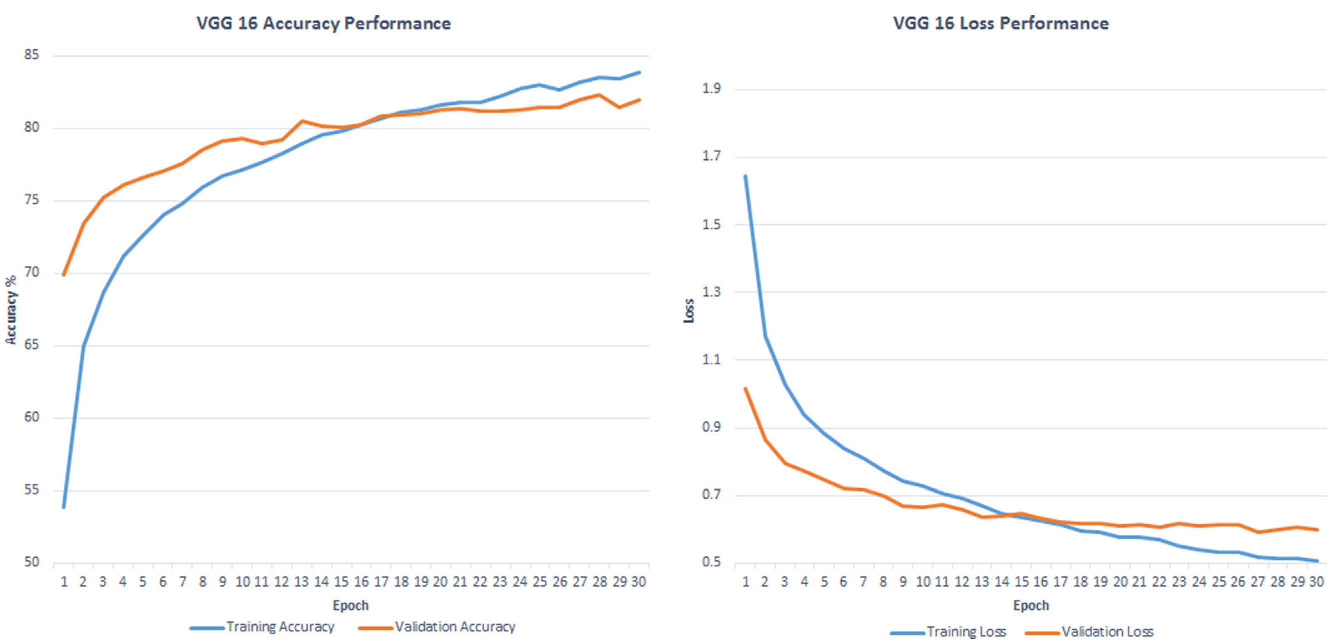

Fig. 3. VGG16 Net, left is accuracy of the model and right depicts the model Loss.

Training and fine-tuning the deep learning architectures is carried out as specified in Section 2.1. The results of the experiments are presented in Figs. 2–7. Each figure depicts the accuracy and the entropy log-loss of each architecture

After fine-tuning, the models using 10 epochs all the models except VGG 16 had accuracy above 90%. Furthermore, even after the 30th training iteration, high accuracy results were obtained with substantially reduced log-loss.

ResNet and DenseNets models consistently perform better than VGG 16 and Inception V4. Additionally, they converge easily as perceived in Figs. 4–7.

The deeper models had better test score as detailed in Table 1.

ResNet 50 and ResNet 101 performs adequately with fewer iterations. On the other hand, ResNet 152 performs poorly with fewer iterations as demonstrated in Figs. 4 and 5. However, ResNet 152 increases its accuracy and reduce its log-loss with increased number of iterations as depicted in Fig. 6.

Overall, DenseNets 121 performed well with the highest accuracy and lowest log-loss while VGG 16 performed poorly with the least accuracy and highest log-loss.

4. Discussion

Deep learning models have dominated in the field of machine learning for image processing. Advancement in deep learning and image processing presents an opportunity to extend the research and application to detection and classification of plant disease using images. Fast and accurate models for plant disease identification are desired so that accurate measures can be applied early. Thus, mitigating the issue of food security.

Recent work in deep learning has demonstrated that deeper models are more accurate and efficient to train (Huang et al., 2016). However,

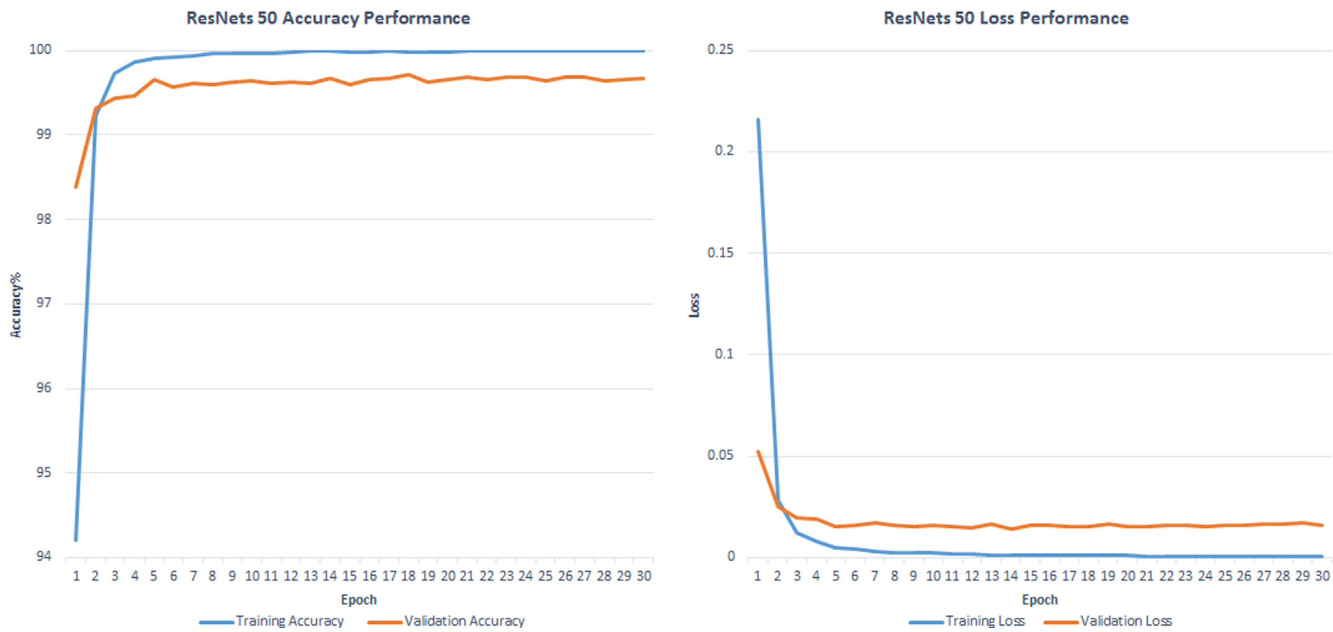

Fig. 4. ResNet with 50 layers, left is Accuracy of the model and Right depicts the model Loss.

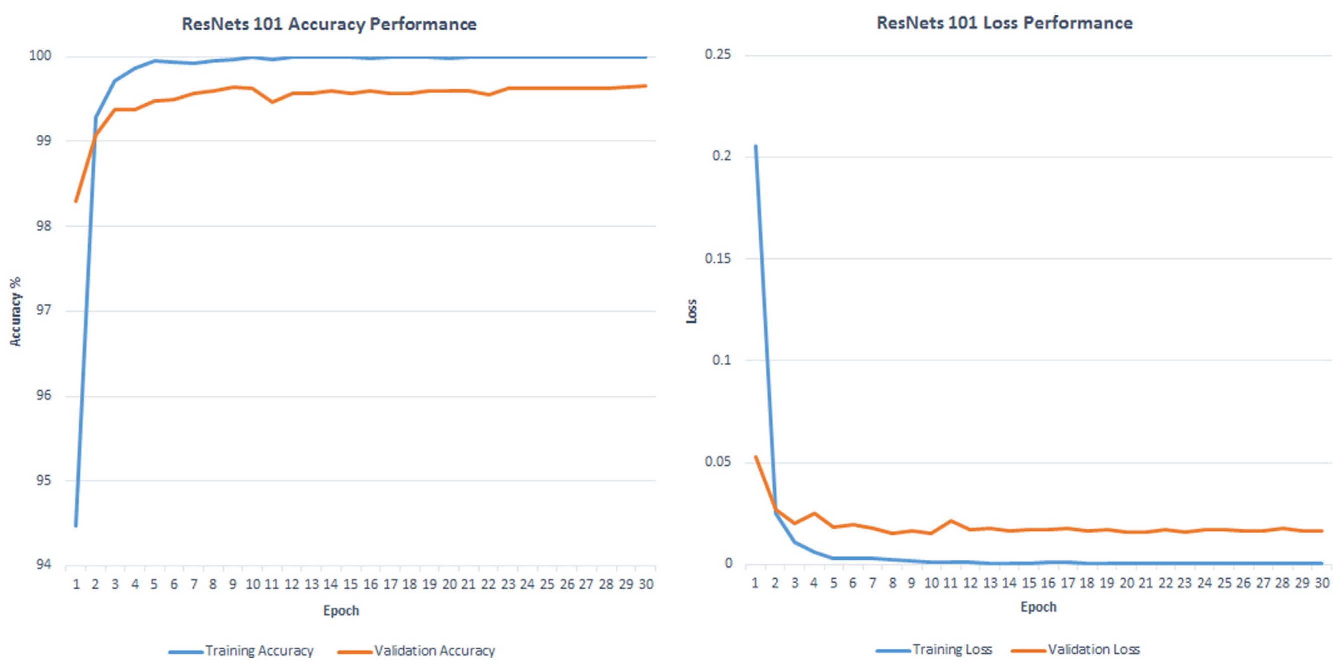

Fig. 5. ResNet with 101 layers, left is accuracy of the model and Right depicts the model Loss.

as the depth increases other challenges emerge such as vanishing gradients, internal covariate shifts and degradation problem. Additionally, there has been computational cost that emanate from training deep models. Strategies to deal with some of these problems have been proposed for different architectures. These including skip connections (He et al., 2016), transfer learning (Pan and Fellow, 2009), initialization strategies (Mishkin and Matas, 2016), optimization methods (Le et al., 2011) and Batch Normalization (Szegedy and Com, 2015).

From literature a number of image processing (Samanta et al., 2012), machine learning (Athanikar and Badar, 2016; Wang et al., 2012) and deep learning (Mohanty et al., 2016; Sladojevic et al., 2016) models have been to applied to area of disease identification using images. With deep learning showing outstanding performance. Transfer learning concept has been adopted (Mohanty et al., 2016) and demonstrates that it aids in boosting accuracy as well as reducing the execution time.

An extension of research by analyzing state-of-the-art deep learning models in the plant disease identification is carried out. Fine-tune and comparative evaluation of VGG net, Inception V4, ResNet (50,101 and 152 layers) and DenseNets is done. These architectures have been successfully applied in different tasks such as ImageNet, Cifar 10 and Cifar 100 classification.

In contrast, DenseNets, ResNet and Inception V4 performed relatively well compared to VGG net as illustrated in the test score on Table 1. This proof that deeper network performs well than shallow networks. Equally, the number of parameters on the deeper networks (DenseNets, ResNet and Inception V4) are reduced compared to VGG net. With DenseNets having the least number of parameters. DenseNets 121 has considerably reduced number of parameters even though its similar to ResNet. DenseNets is 8 times less than ResNet 152 and 16

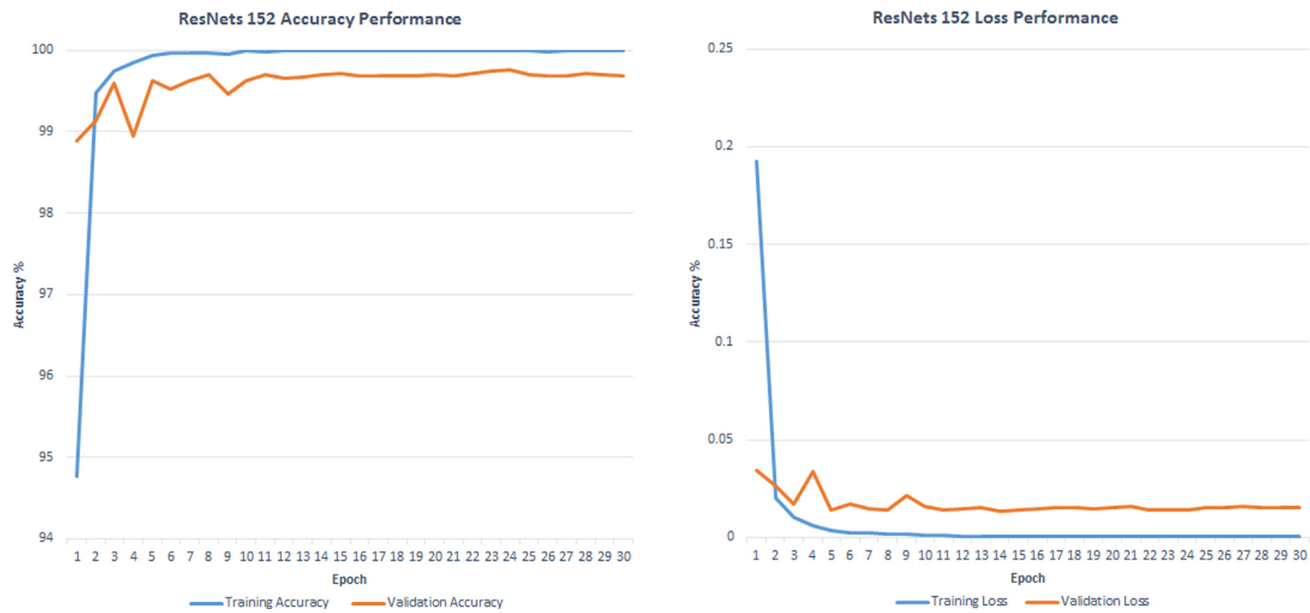

Fig. 6. ResNet with 152 layers, left is accuracy of the model and Right depicts the model Loss.

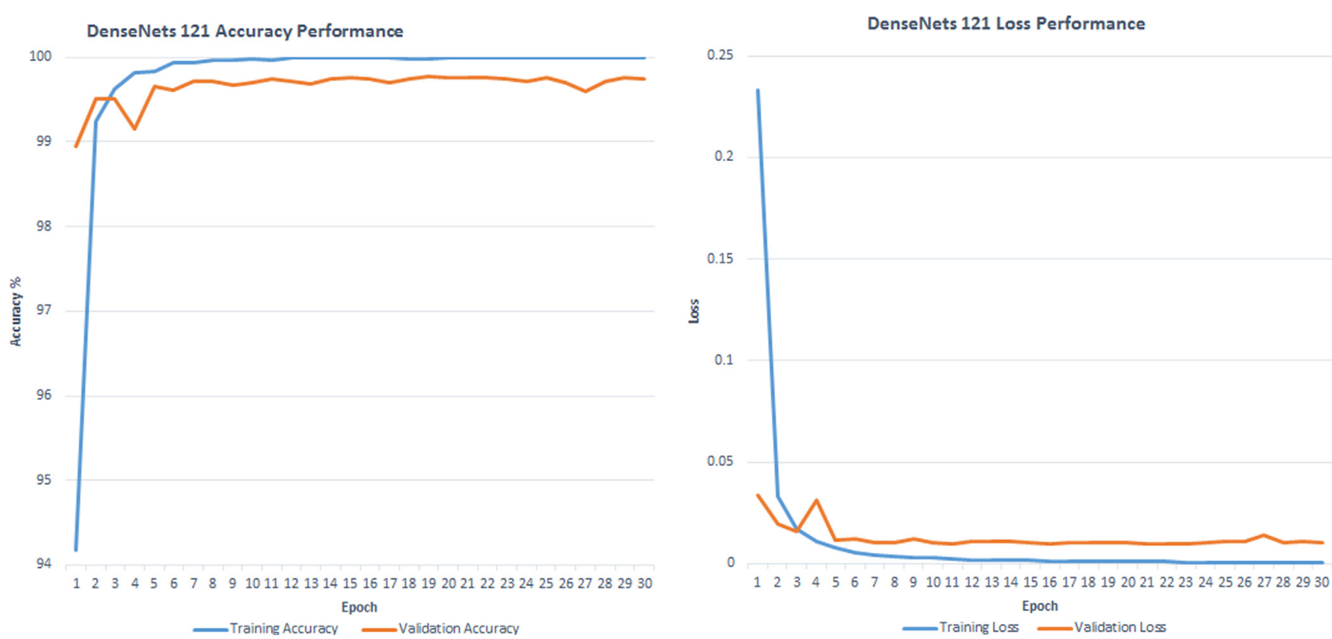

Fig. 7. DenseNet with 121 layers, left is accuracy of the model and Right depicts the model Loss.

times less than VGG net. Therefore, it's easier to train DenseNets compared to the rest of the architectures studied. ResNet on the other hand performs well although it takes a lengthy time to train compared to DenseNets. Similarly, Inception V4 is computationally expensive in terms of running time. Inception V4 and VGG net have another challenge regarding the convergence. DenseNets and ResNet architecture demonstrate that extremely deep networks can be more accurate, in addition to requiring less weights.

4.1. Conclusion

In this task, fine-tuning and evaluation of state-of-the-art deep convolutional neural network for image-based plant disease classification is performed. The architectures evaluated include VGG 16, Inception V4, ResNet with 50,101 and 152 layers and DenseNets with 121 layers. From the experiment, DenseNets has a tendency to yield coherent increment in accuracy with rising number of epochs, with no manifestations of performance deterioration and overfitting. In addition, DenseNets requires significantly less number parameters and sensible computing time to accomplish best in classification exhibitions. DenseNets obtains a test accuracy score of 99.75% for the 30th epoch, beating the rest of the architectures. DenseNets is, therefore, a good architecture for the task of plants image-based disease identification. Even though the performance of the architecture is good, further research needs to be done to improve on the computational time.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 61175004; the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant 20121103110029; and National Natural Science Foundation of

China under Grant 31400426.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.compag.2018.03.032.

References

Athanikar, G., Badar, M.P., 2016. Potato Leaf Diseases Detection and Classification System, 5(2), 76–88.

Chetlur, S. et al., 2014. cuDNN: Efficient Primitives for Deep Learning, pp. 1–9.

Deng, L., Yu, D., 2014. Deep Learning: Methods and Applications. Foundations and Trends® in Signal Processing. pp. 3–4.

Dyrmann, M., Karstoft, H., Midtiby, H.S., 2016. Plant species classification using deep convolutional neural network. Biosyst. Eng. 151 (2005), 72–80.

Glorot, X., Bordes, A., Bengio, Y., 2011. Deep sparse rectifier neural networks. AISTATS '11 Proc. 14th Int. Conf. Artif. Intell. Stat., vol. 15, pp. 315–323, 2011.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep Residual Learning for Image Recognition, Arxiv.Org, vol. 7, no. 3, pp. 171–180.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Identity Mappings in Deep Residual Networks. pp. 1–15.

http://opencv.org/.

https://github.com/liuzhuang13/DenseNet.

https://keras.io/.

Huang, G., Weinberger, K.Q., Van Der Maaten, L., 2016. Densely Connected Convolutional Networks.

Hughes, D., Salathe, Marcel, 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics, pp. 1–13.

Krizhevsky, A., Sutskever, I., Geoffrey, E.H., 2012. ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25 (NIPS2012), 1–9. http://dx.doi.org/10.1109/5.726791.

Le, Q.V., Coates, A., Prochnow, B., Ng, A.Y., 2011. On Optimization Methods for Deep Learning,. In: Proc. 28th Int. Conf. Mach. Learn., pp. 265–272.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-Based Learning Applied to Document Recognition. Proc. Of the IEEE.

Melorose, J., Perroy, R., Careas, S., 2015. World population prospects, United Nations 1(6042), 587–592.

Mishkin, D., Matas, J., 2016. All you need is a good init, pp. 1–13.

Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7 (September), 1–7.

Pan, S.J., Fellow, Q.Y., 2009. A Survey on Transfer Learning, pp. 1–15.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Nips, pp. 1–10.

Reyes, A.K., Caicedo, J.C., Camargo, J.E., 2015. Fine-tuning deep convolutional networks for plant recognition. CEUR Workshop Proc. 1391.

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., Mccool, C., 2016. DeepFruits : A Fruit Detection System Using Deep Neural Networks.

Samanta, D., Chaudhury, P.P., Ghosh, A., 2012. Scab Diseases Detection of Potato using Image Processing 3(April), 109–113.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. Int. Conf. Learn. Represent. 1–14.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D., 2016. Deep neural networks based recognition of plant diseases by leaf image classification. Comput. Intell. Neurosci. 2016 (June).

Szegedy, C., Com, S.G., 2015. Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift, vol. 37.

Szegedy, C., Vanhoucke, V., Shlens, J., 2015. Rethinking the Inception Architecture for Computer Vision.

Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A., 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.

Wang, H., Li, G., Ma, Z., Li, X., 2012. Application of neural networks to image recognition of plant diseases. 2012 Int. Conf. Syst. Informatics, ICSAI 2012, no. Icsai, pp. 2159–2164.

Yu, D., et al., Deep convolutional neural networks with layer-wise context expansion and attention. In: Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, vol. 08–12–Sept, 2016, pp. 17–21.