SN Computer Science (2021) 2:297
https://doi.org/10.1007/s42979-021-00691-9

ORIGINAL RESEARCH

High Efficiency Disease Detection for Potato Leaf with Convolutional Neural Network

Trong-Yen Lee[1] · I-An Lin[1] · Jui-Yuan Yu[1] · Jing-min Yang[1] · Yu-Chun Chang[1]

Received: 24 November 2020 / Accepted: 10 May 2021

Abstract
The potato is the fourth largest food crop and is grown in many places. Potato crops are mainly infected with fungi so they suffer from early blight diseases and late blight diseases. Crop diseases must be detected and recognized because plant diseases have a significant effect on production. Smart farming using machine learning allows infected crops to be identified automatically. Real-time control of disease and management increases productivity and reduces agricultural losses. This study proposes a highly efficient CNN (convolutional neural network) architecture that is suitable for potato disease detection. A database is created for the training set using image processing. Adam is used as the optimizer and cross-entropy is used for model analysis. Softmax is used as the final judgment function. The convolution layer and resources are minimized but accuracy is maintained. The experimental results show that the proposed model detects plant disease with 99.53% accuracy and reduces parameter usage by an average of 99.39%.

Keywords Smart farming · Machine learning · Convolutional neural network (CNN) · Disease detection

Introduction

Early identification of crop diseases depends on professionals observing the symptoms with the naked eye, so the disease causes a certain degree of damage in the field and it is not easy to observe in large areas of dedicated cultivation.

AI has found many applications in the farming industry. Urban development reduces the amount of land that can be used to feed a growing population. Plant diseases have a significant effect on the production of crops and result in major economic losses in agriculture. Neural networks can be used to accurately identify plant states in an intelligent and automated manner so machine learning can be used to detect crop diseases and reduce agricultural losses and increase quality and yield.

This study proposes a highly efficient CNN architecture to detect leaf disease and the sources of plant datasets are potatoes. The potato is one of the most consumed crops. Early blight and late blight are the most common diseases in potatoes. This study determines the best model and compares the results in terms of accuracy and parameters.

Related Work

Image processing techniques are widely used in agriculture. Machine learning can be used to identify plant leaf diseases to achieve smart farming and studies propose various techniques for plant disease recognition.

Most studies use a support vector machine (SVM) as the core of disease recognition to detect leaf disease. This is the supervised machine learning algorithm, which is used for classification and regression. Kaur et al. [1] proposed a

This article is part of the topical collection "Applications of Cloud Computing, Data Analytics and Building Secure Networks" guest-edited by Rajnish Sharma, Pao-Ann Hsiung and Sagar Juneja.

✉ Trong-Yen Lee
tylee@ntut.edu.tw

I-An Lin
cabade7167@gmail.com

Jui-Yuan Yu
richyu220@hotmail.com

Jing-min Yang
yjm4201@163.com

Yu-Chun Chang
marta3252858@gmail.com

[1] Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan

Published online: 26 May 2021

segmentation approach and used a support vector machine to demonstrate disease classification in over 300 images. In [2], an image segmentation with a multiclass SVM is proposed by Islam et al. In [3], Md. Selim Hossain et al. propose an image processing system for tea plant disease recognition using a support vector machine. Prakash et al. [4] proposed a method for the detection of diseases in citrus leaves and the leaves are classified as diseased or not using a SVM. In [5], Iqbal et al. propose a system to detect potato leaf diseases using a Random Forest classifier, which performs better than related works [3, 3] which are SVM-based methods.

The development of neural class means that many studies use reel neural networks, instead of an SVM. In [7], an AutoEnconder CNN method is proposed by Pardede et al. In [8], Tm et al. propose an automatic feature extraction method for the neural network model, which allows more than 90% accuracy.

There are many CNN models, such as AlexNet, VGG16, and Inception. One study [9] evaluated the detection of almond disease, using deep learning models, such as AlexNet, VGG16 and VGG19 and showed that VGG16 is the most accurate. In [10], Suryawati et al. evaluated the detection of tomato diseases using AlexNet, VGG16, and GoogLeNet and showed that VGG16 gives the most accurate results.

VGG16 is relatively effective for the detection of leaf diseases so the architecture of the proposed method used VGG16 to implement machine learning for plant disease detection. A previous study by the authors proposed a detection system for farming that uses a convolutional neural network (CNN) in the artificial neural network to determine symptoms in plant leaves [11]. This study uses the proposed CNN architecture, model selection and compares in more detail to determine how to improve accuracy and resource utilization.

Early Blight and Late Blight in Potatoes

Early blight and late blight hamper potato production. The early disease occurs before the potato flowers, and late blight usually occurs before the harvest of young tubers, as shown in Fig. 1.

When potato leaves become infected with early blight, dark brown and gray-white spots that are approximately circular appear and become irregular if the spots expand to the veins of the leaves. If it is too wet, a black mold layer is found on the spot. When potato leaves become more diseased, they bind to each other into larger areas of dead spots, causing the leaves to wilt in part or in whole. Early blight mainly affects leaves, stems, and tubers. When the shanks and stems become infected, long bar-shaped brown spots are formed.

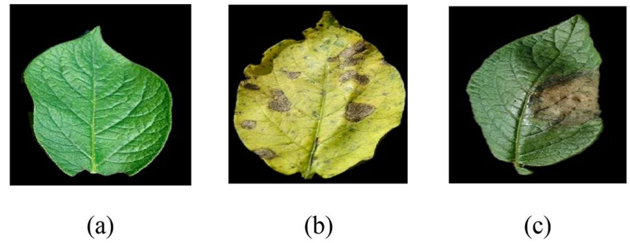

Fig. 1 Potato leaves: **a** healthy, **b** early blight and **c** late blight

If potato leaves are infected with late blight, the initial spots are water-stained, usually starting in the sharp corners of the leaves and at the leaf edge. When the environment is humid expansion is accelerated. Disease spots and healthy parts are not clearly separated. In the area around the disease, spots have a sparse white mold and the disease expands to the main vein and leaf handle so the leaves wilt and droop and eventually, the whole plant becomes charred black, showing wet rot.

Dataset Description

Data Augmentation

Data augmentation is required for the proposed system because deep learning requires much data. The dataset was obtained from an Open Access image database PlantVillage. The total number of leaves is 2152 images: 152 images of healthy potato leaves, 1000 images of early blight leaves and 1000 images of late blight leaves. The Keras image preprocessing module provides an ImageDataGenerator class that is capable of applying augmentation methods [12]. The processing for data augmentation involves random scaling and horizontal and vertical translation, which prevents overfitting. The original images are processed by horizontal movement, vertical movement, zoom in and out, horizontal flipping, and vertical flipping.

The total dataset is 12912 pictures and the training picture data has 11621. The test set data comprises 1,291 random sheets, after random horizontal movement, random vertical movement, random zoom in and out, random horizontal flipping, and random vertical flipping. An example of an image database is shown in Fig. 2. This is used to compare the results and accuracy and to observe the trained results.

Image Pre-processing

To allow picture training based on the color, image preprocessing does not allow other colors to interfere. Gaussian filtering removes noise, the picture is normalized to reduce the effect of the light and the formula is used to extract

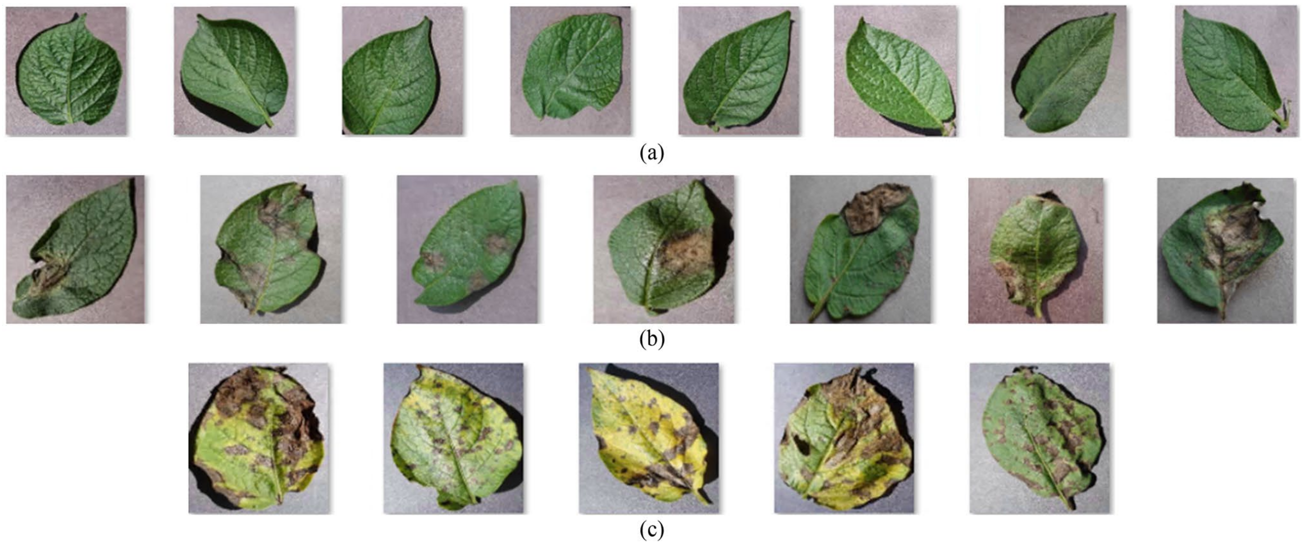

Fig. 2 Example of an image database for potato leaves: **a** healthy **b** early blight and **c** late blight

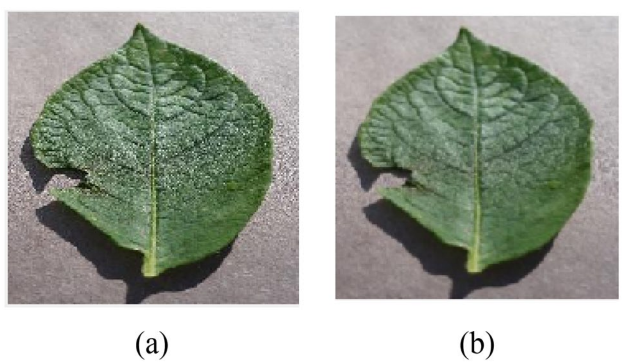

Fig. 3 Potato leaves: **a** original picture and **b** 7×7 Gaussian yields with $\sigma = 0.8$

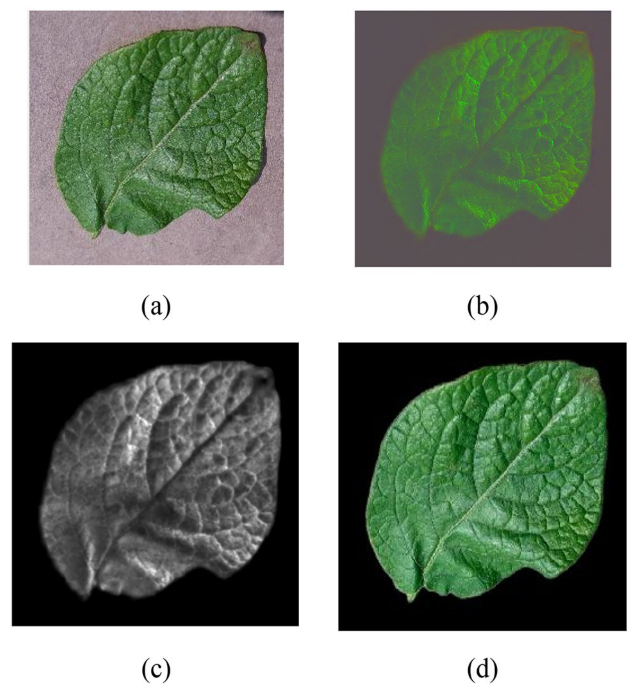

Fig. 4 Potato leaves: **a** original picture, **b** after normalization, **c** after green fortification and **d** processed image of a potato leaf

the desired area, remove the scattered picture points and then restore the reconstructed picture. All of these steps are shown in Figs. 3 and 4.

- Gaussian filtering
  Gaussian filtering removes noise from the pictures. An original picture of a potato leaf and the result of Gaussian filtering are respectively shown in Fig. 3a, b. For the Gaussian filter, σ is the standard deviation [13]. If σ is too large, the picture is blurred and $\sigma = 0.8$ gives good edge smoothing. The experiments use a 7×7 Gaussian filter and a standard deviation of 0.8.
- Picture normalization
  To accelerate the solution for machine learning, data are normalized. Normalization in machine learning is data normalization, so data converge quickly accelerates the process of finding solutions. Without normalization, there are different dimensions when the gradient decreases and rates are different. With data normalization, this step gives dimensions of the same order of magnitude and accelerates convergence.
  An RGB color model is a commonly used format for image processing but this is susceptible to changes in light and shadow. Therefore, during the processing of the image, RGB is normalized to eliminate the effects of illumination. RGB normalization uses Eq. (1), where R', G',

and *B'* are the color of RGB after normalization. In this step, red and green are significantly less affected by color, brightness and the light source. To demonstrate RGB normalization, an original picture is shown in Fig. 4a, in which the shadow at the edge of the leaf disappears, as shown in Fig. 4b.

$$R' = \frac{R}{R+G+B} . G' = \frac{G}{R+G+B} . B' = \frac{B}{R+G+B} \tag{1}$$

- Region of interest

  The region of interest for the proposed method is the green area. To obtain information about green pixels, out the green area is removed using Eq. (2). If F is greater than 0, green items are presented. If there is more green color than red or blue, the value for RGB is greater than 0 so the value of the green area is greater than 0. As shown in Fig. 4c, the parts other than green turn black.

$$F = G' + G' - R' - B' \tag{2}$$

- Color Conversion

The picture is open and closed to remove the surrounding noise. The image is restored to the original image, as shown in Fig. 4a, and is used as to reconstruct, as shown in Fig. 4d.

Proposed CNN System Architecture

Deep learning is a machine learning algorithm, which is used for image processing, image restoration, and speech recognition. The word "deep" implies that deep learning has more layers than machine learning and that each layer is closely related, so the output of the previous layer is used as the input of the next layer. Deep learning does not specifically mark and then extract features. When training the model, the model automatically extracts features and optimizes the learning algorithm and the matching data. The process of learning can be unsupervised, supervised, or semi-supervised.

The CNN has four layers: a convolution layer, a pooling layer, an activation function layer, and a full connection layer, as shown in Fig. 5. The CNN gives good image classification and object identification. In the experiment involving plant disease recognition, a convolutional neural network (CNN) is used for deep learning.

Pooling Layer and Overfitting

Pooling is a nonlinear form of down-sampling. The process of pooling reduces the input for a zone to a single value.

For convolutional neural networks, this information set provides information that is similar to the output connection while reducing memory consumption. Maximum pooling divides the image into several square blocks and extracts the maximum value for the output for all the blocks. The largest feature of the area that is removed from pooled layer reduces the size of the data each time. Pooling provides basic invariance for rotation and translation and increases the accuracy of object detection for convolutional networks.

Drop out must be applied to part of the pooling layer. Drop out randomly allows some neurons, which are nodes in the class of nerves, to drop out from the corresponding input and output. This is not necessary but prevents overtraining (overfitting). If this model is not overtrained (overfitted), training can continue to reduce loss until accuracy reaches 100%. However, in reality, it is impossible to achieve 100% accuracy stably so over-training (overfitting) occurs.

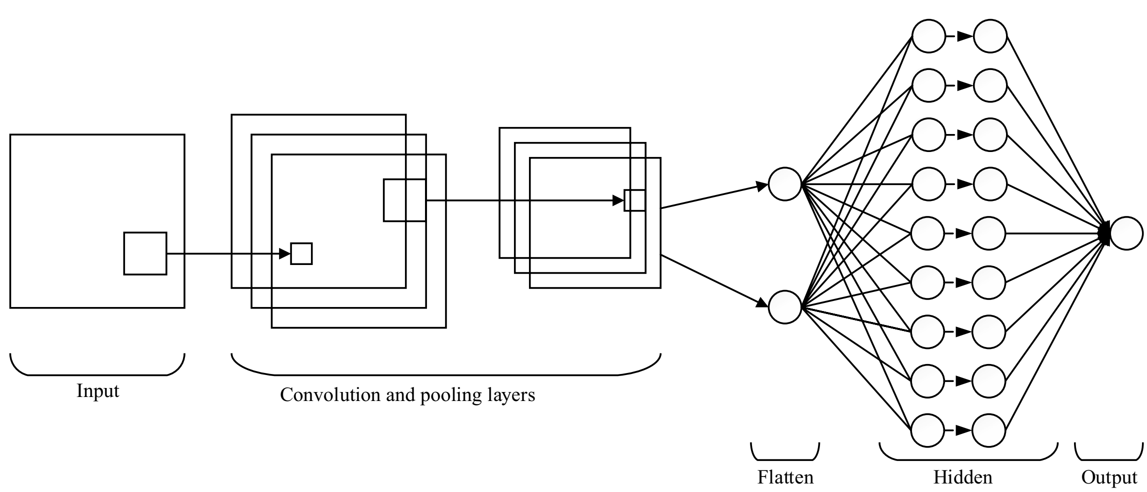

Fig. 5 The CNN system architecture

The model can be trained to increase the number of training cycles. The goal is to increase and maintain the highest level.

Activation Function

The activation function allows a CNN to learn complex models, such as images, videos, and sounds. A Rectified Linear Unit (ReLU) is a common activation function that is composed of an identity function and a threshold activation function, so the gradient does not disappear. The proposed system uses image recognition so the ReLU is used to prevent gradient disappearance.

The parameter $f(a)$ is the activation function and solves nonlinear problems using neural networks. Eq. (3) of a is the value of the neural network node. In Eq. (4), $W_{i,j}$ is weight, b_j is the bias value and x_j is x the j pixel value.

$$f(a) = \begin{cases} a, a \geq 0 \\ 0, a < 0 \end{cases} \tag{3}$$

$$a_i = \sum_j W_{i,j} x_j + b_i \tag{4}$$

Fully Connected Layers

The fully connected (FC) layers follow decision functions. A Softmax [14] function is used for the proposed system as a decision function to determine the probability of potato leaf disease. The Softmax function is shown as Eq. (5). All values are between 0 and 1, which is consistent with the probability. The number K is the final output. Equation (5) for a gives the value of the neural network node. σ is the value of Softmax. The result for the proposed method is the condition of the plant: whether it is sick. Therefore, the Softmax function is used.

$$\sigma(a)_i = \frac{e^{a_i}}{\sum_j e^{a_i}}, i = 1, \ldots, K \tag{5}$$

Convolutional Layer Scanning

In the convolutional layer, a series of mathematical operations are performed to extract and the input image is reduced to a smaller size using a filter. Starting from the upper left corner, as shown in Fig. 6, if the black points are individual pixels using a 3×3 filter, then after convolution, the filter removes the largest feature part of nine pixels and combines these into a new pixel. The filter starts to move and scan step by step until the process ends. A new matrix that is smaller than the input image is generated.

Each filter gives a new value. This value is a feature that is related to the image, which may be the tip of the leaf or the withered part of the leaf. The CNN extracts and learns these features. A filter has a definite length, width, and height. If the image is black and white, then the depth is 1. The leaf image that is used is a color image that is composed of RGB, so the depth of the filter is 3. The filter compresses the image and retains the characteristic value.

Convolution involves progressive scanning so a CNN extracts the features of the leaves, regardless of the

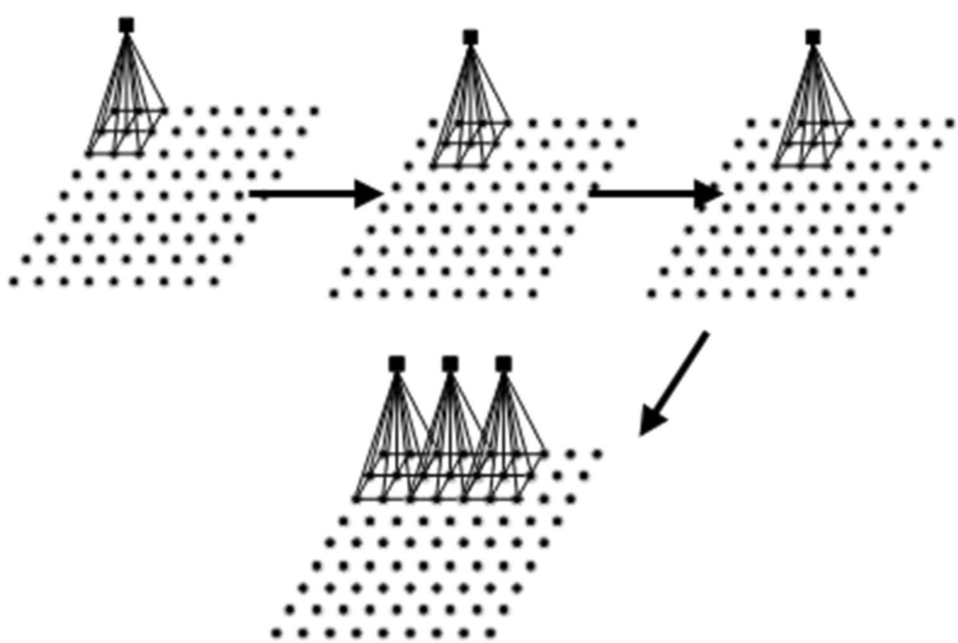

Fig. 6 Diagram of filter scanning

symptoms that the leaves exhibitor where the leaves are in the picture. Scanning continues step repeatedly until new pixels fill the entire image.

Experimental Results

The proposed method is implemented on a Windows10 machine running an Intel®Core(TM) i7-7700K CPU @4.2GHz (8CPUs),4.5GHz with Jupyter [15] environment and Python 3.5 kits and the Keras suite is used for software development. This study uses three models with different depth and width: A, B and C. The model that has the highest accuracy and the lowest loss rate is used to construct the framework. Finally, precision and parameters are compared for VGG16 and VGG19.

Training Process

The process of training is related to the loss rate function. The Mean Square Error (MSE) and the Mean Absolute Error (MAE) are common loss rate functions for data processing. In general, MAE is better for data anomalies (specific points) than MSE and MSE is better than MAE if there are fewer special points.

The loss rate function for this study is shown as Eq. (6). This is a cross-entropy. The number of samples is n and z is the number of labels. Cross entropy gives good classification and is used in the role function. The Adam (Adaptive Moment Estimation) method is used for gradient-based optimization algorithms for deep neural networks [16], so Adam is implemented in the proposed method. The Softmax function is used as the decision function. The value of $y_{i,t}$ is the output from Softmax. The smaller the difference between the predicted and the actual values, the smaller is the value of LOSS.

$$\mathrm{LOSS} = -\sum_{i=1}^{n}\sum_{t=1}^{z}(y_{i,t} * \log(y_{i,t})) \tag{6}$$

The loss rate curves for the training set and the test set for 1–20 cycles and 1–100 cycles are shown in Fig. 7. The blue line is the loss rate curve for the validation set and the orange line is the loss rate curve for the test set. The fastest convergence is between the number of training 1–13. This area has the highest efficiency. The data is unstable if there are 30–40 training cycles so the training may not be complete and the error is large. The curve is stable if there are more than 40 training cycles, so this model is used.

In ideal conditions, the LOSS for the model test and training decreases as the number of training cycles increases and eventually becomes zero. In reality, the LOSS for test and training is parallel as the number of training cycles

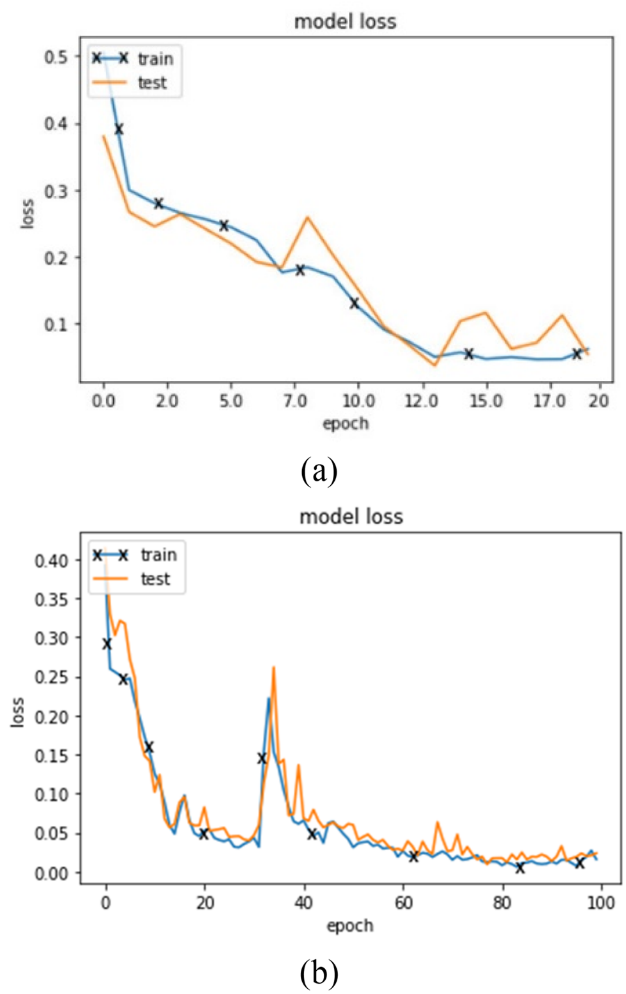

Fig. 7 Diagram of loss rate: **a** for 20 cycles and **b** for 100 cycles

increases. If there are more than 40 cycles, the loss rate for the proposed model is between 0.01 and 0.4. The loss rate remains within this range after further training.

The results for the proposed method are validated to determine whether potato leaves are infected and whether there is early blight or late blight. The proposed method determines whether potato leaves are infected whether there is early blight or late blight.

The results for the epoch data are shown in Fig. 8. The training curve has an extremely large jitter in the beginning, as shown in Fig. 8a. For 100 training sessions, the result for the epoch data gradually levels but does not converge, as shown in Fig. 8b. If the number of training sessions is increased to 300, the training curve gradually converges after about 200 training sessions and there is no overfitting, Fig. 8c.

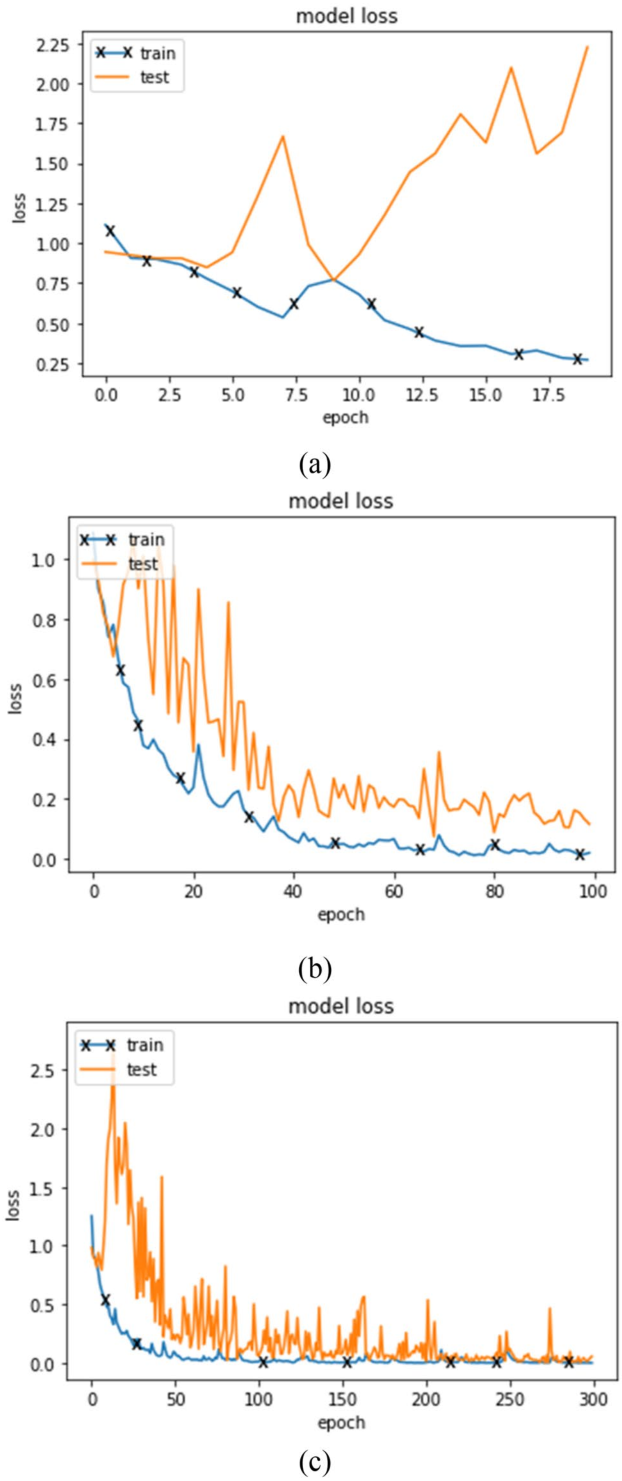

Fig. 8 The results for the epochs: number of training sessions—**a** 1 ~ 20 **b** 1 ~ 100 **c** 1 ~ 300

Table 1 Parameters of model A

Neural layer	Number of nodes
Convolutional layer × 2	64
Pooling layer	64
Dropout	64
Convolutional layer × 2	128
Pooling layer	128
Dropout	128
Convolutional layer × 4	256
Pooling layer	256
Dropout	256
Convolutional layer × 4	512
Pooling layer	512
Dropout	512
Fully Connected Layer × 2	512

Table 2 Parameters of model B

Neural layer	Number of nodes
Convolutional layer	64
Pooling layer	64
Dropout	64
Convolutional layer × 2	128
Pooling layer	128
Dropout	128
Convolutional layer × 2	256
Pooling layer	256
Dropout	256
Fully connected layer	256

Table 3 Parameters of model C

Neural layer	Number of nodes
Convolutional layer	64
Pooling layer	64
Dropout	64
Convolutional layer × 2	128
Pooling layer	128
Dropout	128
Convolutional layer × 3	256
Pooling layer	256
Dropout	256
Fully connected layer	512

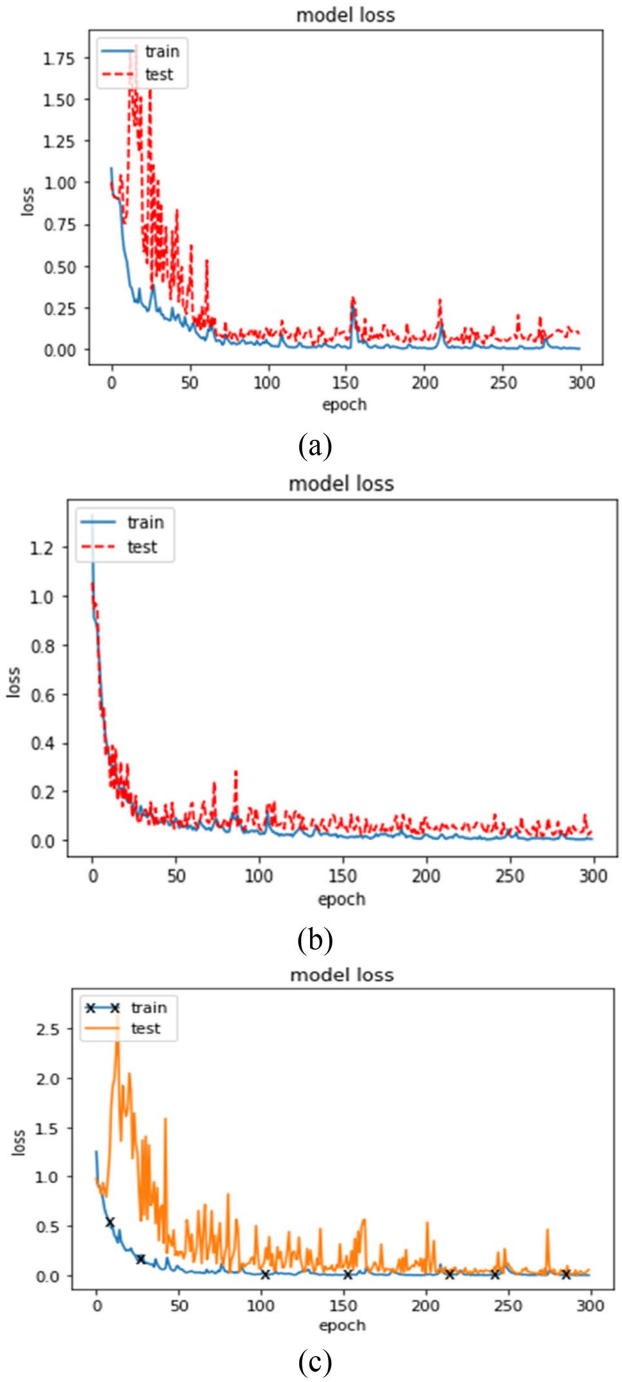

Fig. 9 The results for the loss values: **a** model A **b** model B and **c** model C

Model Architecture and Comparison

Three models with a different width and depth are used: A, B and C. 300 training cycles are used to determine which model is best for plant disease recognition. In this experiment, the parameter settings for each model are shown in Tables 1, 2 and 3. Model A is the deepest and model B is the shallowest. This setting determines the depth of the model that matches the data.

The results for the loss values for each model are shown in Fig. 9. In Fig. 9a, there is slight overfitting for 75 training cycles so the model is too deep. Fig. 9b shows the loss for model B. The curve gradually stabilizes after about 50 training cycles and the accuracy reaches a maximum, so this model that can be used. Fig. 9c shows the loss for model C. The curve gradually stabilizes after about 150 training cycles and the accuracy is greater so this model can be used. The results show that Model A is too deep and architectural model B is more accurate than model C. Model C is the most accurate, with the lowest loss rate so the proposed method uses model C to construct the framework.

Proposed CNN

This study proposes a CNN configuration for blade detection, as shown in Fig. 10. The detail of parameter utilization is shown in Table 4. There are not enough types and quantities of the materials for this proposed system so the model reduces the number of nodes before and after the model and the middle part is maintained, which reduces the likelihood of overfitting and the time that is required to train because fewer resource parameters are used. In the pooling layer, there is no dropout but this prevents overfitting. Ideally, only this model is required.

If there is no overfitting, training continues until the loss rate loss is reduced and the accuracy reaches 100%, but it is impossible to achieve an accuracy of 100%. Overfitting is inevitable so this study increases the number of training cycles for the model and maintains accuracy.

The results for the training set are shown in Fig. 11. The initial training curve for LOSS is very unstable, as shown in Fig. 11a. If there is a large shock, the training curve gradually becomes stable after 100 cycles, but there is no convergence. For 300 training cycles, the number of training cycles stabilizes after about 200 cycles and the line gradually converges, with no overfitting. The accuracy of disease recognition is 99%, as shown in Fig. 11b.

Comparison

The experimental model was implemented using the VGGNet family architecture model [17]: VGG19 and

Fig. 10 The proposed CNN

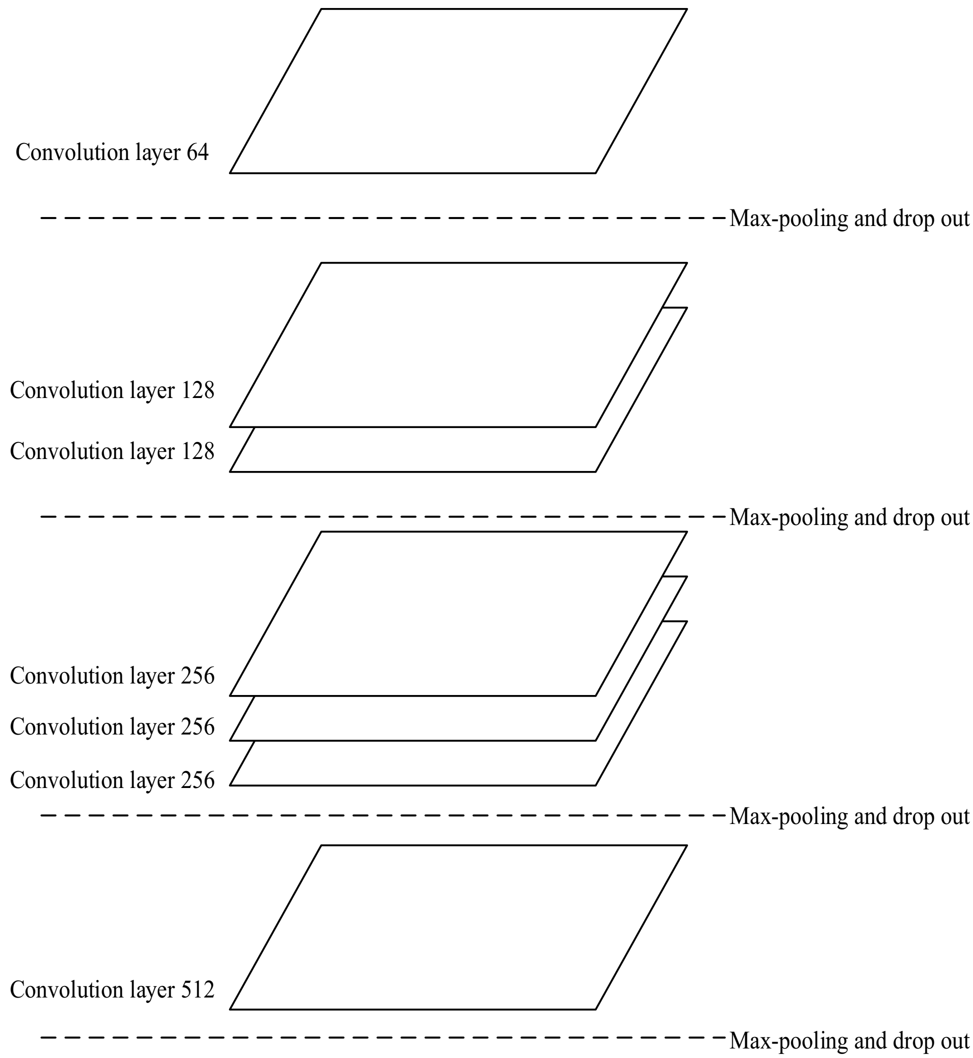

Table 4 Parameters for the proposed CNN

Neural layer	Number of nodes	Parameters
Convolutional layer	64	1792
Pooling layer	64	0
Dropout	64	0
Convolutional layer	128	73,856
Convolutional layer	128	147,584
Pooling layer	128	0
Dropout	128	0
Convolutional layer	256	295,168
Convolutional layer	256	590,080
Convolutional layer	256	590,080
Pooling layer	256	0
Dropout	256	0
Fully Connected Layer	512	8,390,659
Total		10,089,219

VGG16. The results are shown in Table 5. If there is only one training session, the VGG19 model performs best in terms of the accuracy rate of prediction, but if the number of training sessions is increased, the accuracy is similar to that for a single training session. The accuracy of the VGG16 model increases significantly as the number of training sessions increases. The proposed method is better than the VGG16 model in terms of convergence and speed.

The prediction accuracy of the proposed method is not initially the best in the initial but if the number of training sessions is increased, the accuracy of prediction is better than that for the VGG19 model and the VGG16 model. The proposed method performs best in terms of parameter utilization, with a 99.39% reduction, so the fewer the number of resources that are used, the faster are calculations executed.

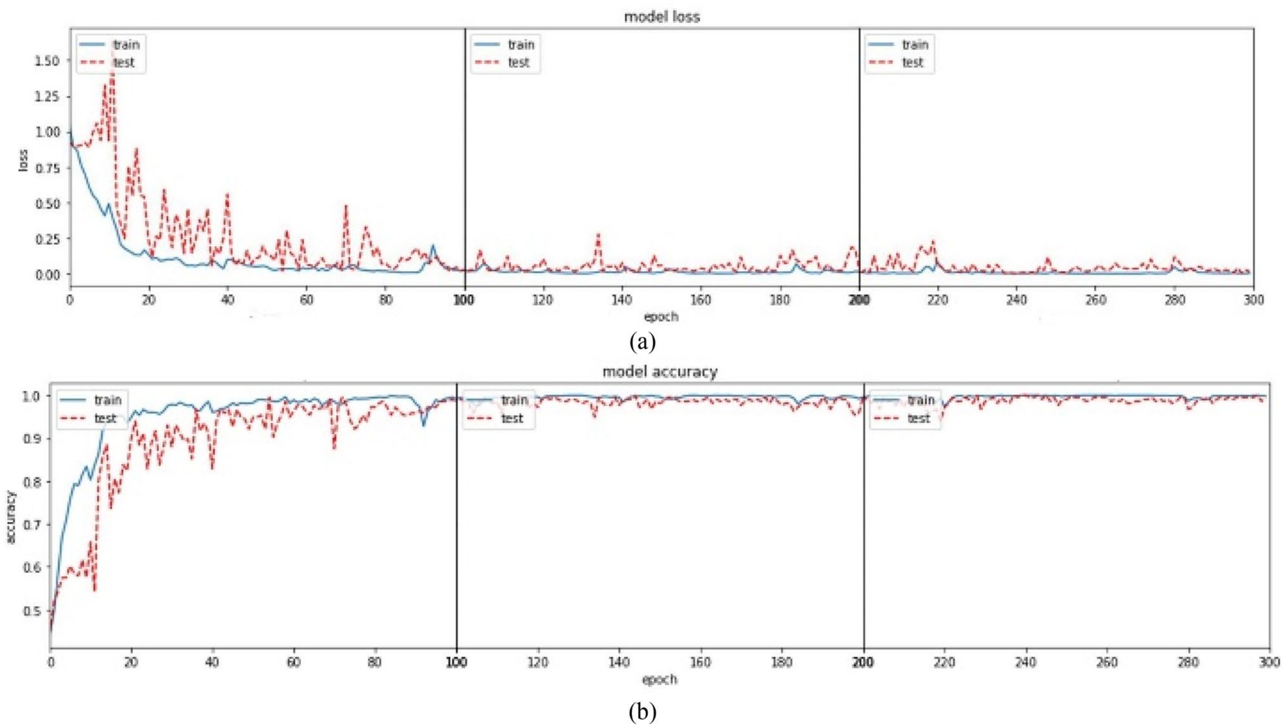

Fig. 11 Results for the training set: **a** loss and **b** accuracy

Conclusion

The proposed method to detect the health status of potato leaves uses the structure of a convolutional neural network for automatic crop disease recognition. Using a large number of image databases and image processing, a representative database is established to increase the accuracy of the convolutional neural network. This study uses a smaller model so fewer resources are used for a small convolutional neural network. The proposed architecture is applicable to the recognition of plant disease. Using a database with more than 2,000 images and less resources, the accuracy of disease recognition is 99%.

Table 5 Comparison of precision and parameters

Model	Precision (%)			Parameters	Reduction (%)
	Training session		Best		
	For 1	For 100			
VGG16	45.37	91.67	98.15	138,357,544	99.07
VGG19	46.69	47.83	48.55	143,667,240	99.72
New proposed	44.91	97.68	99.53	10,089,219	99.39 (average)

Funding This study was funded by the Ministry of Science and Technology of the Republic of China, Taiwan, for partially supporting this research under Contract No. MOST 108-2221-E-027-091.

Declaration

Conflict of interest On behalf of all authors, the corresponding author declares that there is no conflict of interest.

References

1. Kaur R, Kang SS. An enhancement in classifier support vector machine to improve plant disease detection. In: Proceedings of the 3rd international conference on MOOCs, innovation and technology in education (MITE), Amritsar, 2015, pp. 135–40.
2. Islam M, Dinh A, Wahid K, Bhowmik P (2017) Detection of potato diseases using image segmentation and multiclass support vector machine. In: Proceedings of Canadian conference on electrical and computer engineering (CCECE), Windsor, 2017, pp. 1–4.
3. Hossain S, Mou RM, Hasan MM, Chakraborty S, Razzak MA. Recognition and detection of tea leaf's diseases using support vector machine. In: Proceedings of 2018 IEEE 14th international colloquium on signal processing and its applications (CSPA), Batu Feringghi, 2018, pp. 150–54.
4. Prakash RM, Saraswathy GP, Ramalakshmi G, Mangaleswari KH, Kaviya T. Detection of leaf diseases and classification using digital image processing. In: Proceedings of international conference on innovations in information, embedded and communication system (CIIECS), Coimbatore, 2017, pp. 1–4.

5. Iqbal MdA, Talukder KH. Detection of potato disease using image segmentation and machine learning. In: Proceedings of 2020 international conference on wireless communications signal processing and networking (WiSPNET), India, 2020, pp. 43–47.
6. Pinki FT, Khatun N, Islam SMM. Content based paddy leaf disease recognition and remedy prediction using support vector machine. In: Proceedings of 2017 20th international conference of computer and information technology (ICCIT), Dhaka, 2017, pp. 1–5.
7. Pardede HF, Suryawati E, Sustika R, Zilvan V. Unsupervised convolutional autoencoder-based feature learning for automatic detection of plant diseases. In: Proceedings of 2018 international conference on computer, control, informatics and its applications (IC3INA), 2018, pp. 158–62.
8. Tm P, Pranathi A, SaiAshritha K, Chittaragi NB, Koolagudi SG. Tomato leaf disease detection using convolutional neural networks. In: Proceedings of 2018 Eleventh international conference on contemporary computing (IC3), Noida, 2018, pp. 1–5.
9. Turkoglu M, Hanbay D. Apricot disease identification based on attributes obtained from deep learning algorithms. In: Proceedings of the international conference on artificial intelligence and data processing (IDAP), Malatya, Turkey, 2018, pp. 1–4.
10. Suryawati E, Sustika R, Yuwana RS, Subekti A, Pardede HF. Deep structured convolutional neural network for tomato diseases detection. In: Proceedings of the international conference on advanced computer science and information systems (ICACSIS), Yogyakarta, 2018, pp. 385–90.
11. Lee TY, Yu JY, Chang YC, Yang JM. Health detection for potato leaf with convolutional neural network. In: Proceedings of international conference on computing, analytics and networks (Indo-Taiwan ICAN), India, 2020, pp. 1–5.
12. Chollet F. Building powerful image classification models using very little data. The Keras Blog, 05/06/2015 (2015). https://blog.keras.io/. Accessed 14 July 2018.
13. Shin DH, Park RH, Yang S, Jung JH. Block-based noise estimation using adaptive Gaussian filtering. IEEE Trans Consum Electron. 2005;51(1):218–26.
14. Hu R, Tian B, Yin S, Wei S. Efficient hardware architecture of softmax layer in deep neural network. In: Proceedings of international conference on digital signal processing (DSP), Shanghai, China, 2018, pp. 1–5.
15. Jupyter. https://jupyter.org/. Accessed 14 May 2020.
16. Kingma DP, Ba J. Adam: a method for stochastic optimization. In: Proceedings of international conference on learning representations (ICLR), San Diego, 2015, pp. 1–15.
17. Simonyan K, Zisserman A.Very deep convolutional networks for large-scale image recognition. In: Proceedings of computational and biological learning society, 2015, pp. 1–14.