Tomato leaf diseases recognition based on deep convolutional neural networks

Kai Tian,[1] Jiefeng Zeng,[2] Tianci Song,[3] Zhuliu Li,[3] Asenso Evans,[4] Jiuhao Li[4]

[1]National Supercomputing Center in Shenzhen, Shenzhen, Guangdong, China; [2]College of Water Conservancy and Civil Engineering, China Agricultural University, Beijing, China; [3]Department of Computer Science and Engineering, University of Minnesota Twin Cities, Minneapolis, Minnesota, United States of America; [4]College of Water Conservancy and Civil Engineering, South China Agricultural University, Guangzhou, China

Abstract

Tomato disease control remains a major challenge in the agriculture sector. Early-stage recognition of these diseases is critical to reduce pesticide usage and mitigate economic losses. While many research works have been inspired by the success of deep learning in computer vision to improve the performance of recognition systems for crop diseases, few of these studies optimised the deep learning models to generalise their findings to practical use in the field. In this work, we proposed a model for identifying tomato leaf diseases based on in-house data and public tomato leaf image databases. Three deep-learning network architectures (VGG16, Inception_v3, and Resnet50) were trained and tested. We packaged the trained model into an Android application named TomatoGuard to identify nine kinds of tomato leaf diseases and healthy tomato leaves. The results showed that TomatoGuard could be adopted as a model for identifying tomato diseases with a 99% test accuracy, showing significantly better performance than APP Plantix, a widely used APP for general-purpose plant disease detection.

Correspondence: Kai Tian, National Supercomputing Center in Shenzhen, Shenzhen, Guangdong, China.
E-mail: tiankai@nsccsz.cn

Key words: image augmented; deep learning; image classification; android application.

Acknowledgements: the authors would like to thank Zifu He, the researcher at the Institute of Plant Protection of the Guangdong Academy of Agricultural Sciences, for providing the strain with the pathogen of Tomato yellow leaf curl virus and Tomato mosaic virus. In addition, we sincerely thank Dr. Rui Kuang for consulting on deep learning and improving the writing of the manuscript.

Contributions: all authors have contributed equally.

Conflict of interest: the authors declare no potential conflict of interest.

Funding: this work is supported by the Key-Area Research and Development Program of Guangdong Province (Grant No. 2021B0101310002) and the National Key Research and Development Program of China (Grant No. 2020YFB0204802).

Availability of data and materials: the research data have been published on Mendeley Data: http://dx.doi.org/10.17632/369cky7n39.1. The source code and Android application of this article can be found online at: https://github.com/KaiTian4756/tomatogurad

Received for publication: 27 April 2022.
Revision received: 5 August 2022.
Accepted for publication: 12 August 2022.

Journal of Agricultural Engineering 2023; LIV:1432
doi:10.4081/jae.2022.1432

Introduction

According to the FAO (The Food and Agriculture Organization of the United Nations, FAOSTAT), China was the largest producer of tomatoes from 2013 to 2017. Diseases in tomato leaves cause major production and economic losses by reducing both the quality and quantity of productivity in the tomato industry. Moreover, crop disease control issues are closely related to the development of sustainable agriculture. In China, crop disease diagnosis still heavily relies on the experiences of farmers or crop producers. Excessive use of pesticides leads to long-term drug resistance of the bacteria, which severely impairs the disease resistance in the crop (Hanssen *et al*., 2010). To reduce the use of pesticides, the timely and accurate detection of tomato leaf disease in the early stages has become a critical and urgent challenge to tackle.

Traditionally, identification of the type and the severity of the plant disease mainly relied on experienced farmers or trained experts (Liu *et al*., 2017; Riley *et al*., 2002). However, the process is not ideal, and difficult to meet the requirements of modern agriculture because of low efficiency, small scope, and poor real-time performance. With the rapid development of machine learning algorithms in computer vision, especially deep artificial neural network technology, it is now possible to perform an automatic, timely, and accurate diagnosis of crop diseases with the support of massive agricultural information data (Wolfert *et al*., 2017).

Researchers (Chung *et al*., 2016; Shrivastava *et al*., 2017; Sabanci *et al*., 2017; Kezhu *et al*., 2014; Romualdo *et al*., 2014) have used Bayesian classifiers, support vector machines, and artificial neural networks to develop several expert diagnosis systems for disease detection in various crops. Intelligently identifying crop diseases has achieved reasonable success in many specific cases. The traditional machine learning methods consist of image preprocessing, feature extraction, and classifier training (Mokhtar *et al*., 2015). Among these three steps, the feature extraction step is the key challenge. Traditionally, features were extracted based on prior knowledge to obtain the image's colour, shape, texture, and other characteristics, such as scale-invariant feature transform and histogram of oriented gradient. Generally, the extracted features do not generalise into new images for universality (Tamaazousti *et al.*, 2020), which is one of the main reasons that the developments currently stay in the laboratory phase and cannot be applied in practice. More recently, the emerging deep Convolution Neural Network (CNN) (Fukushima *et al.*, 1983; LeCun *et al.*, 1989), generated via reference to the structure of the visual system has performed very well in image feature extraction. Researchers have achieved excellent performances in various types of image recognition tasks, such as face recognition (Parkhi *et al.*, 2015), human eye detection (Liu *et al.*, 2015), pedestrian detection (Tomè *et al.*, 2016), license plate recognition (Masood *et al.*, 2017). Due to the impressive success in image-based recognition, CNN is now widely used to identify plant and crop disease images. Prasad *et al.* (2017a, b) proved that the features captured from CNN were effective in plant species recognition. Yang *et al*. (2017) proposed a rice disease identification method based on deep CNN techniques to identify 10 common rice diseases where their model achieved an average accuracy of 95.48%. Kawasaki *et al.* (2015) presented a cucumber leaf disease detection system based on CNN. Their system achieved an average accuracy of 94.9% in classifying cucumbers into two typical diseased classes and a healthy class. Sladojevic *et al*. (2016) proposed a novel approach based on deep CNN, which can detect 13 different common types of plant diseases by distinguishing the plant leaves from their surroundings, and the average test accuracy was 96.3%. Mohanty *et al*. (2016) developed a CNN-based model to detect 26 diseases in 14 crop species and achieved an accuracy of 99.35%.

Additionally, Fuentes *et al.* (2017) presented a deep-learning-based approach to detect diseases and pests in tomato plants, and the mean average precision for their whole system showed a performance of more than 80% for the best cases, but they only used their private dataset from areas of the Korean Peninsula and did not provide an open source. Generally, most of these studies are mainly limited to their curated data and have not yet demonstrated the application in practice. In addition, while these studies proved that CNN is effective in recognising crop and plant diseases, the CNN-based models could be further tuned and optimised to identify crop and plant diseases in real-life practice. Finally, it has to note that even if massive agricultural information data can be acquired with the support of automatic sensors, the data annotation still relies on manual labelling.

This work aimed to provide a solution for early control of tomato diseases from the perspective of image recognition. For this purpose, we have comprehensively considered data acquisition, model optimisation, result analysis, and application deployment. The specific contributions of this paper are as follows: i) we curated a comprehensive dataset with images from our in-house and other public datasets to improve the diversity of training data; ii) we developed a more generalised CNN-based model. We tuned the classification structure, added batch normalisation and dropout layer into the network, and replaced the flattened layer with a Global max pooling layer; iii) we also explored the mechanism of the CNN model diagnosing tomato leaf diseases by saliency maps and activation maximisation of the Softmax layer; iv) we also implemented the proposed deep learning model in the TomatoGuard Android application for deployment on Android phones.

Materials and Methods

Identification and control methodology of various diseases of tomato leaves

Tomato diseases are caused by various factors including fungi, bacteria, virus, mite (Chowdhury *et al*., 2021). Fungi are the predominant plant pathogens, and can cause multiple diseases, including early blight, septoria leaf spot, target spot, and leaf mould. The leaf symptoms of early blight are large irregular patches of black, necrotic tissue surrounded by larger yellow areas. The leaf spots have a characteristic concentric banding appearance (oyster-shell or bull's eye). The effective control method minimises leaves' wetting by using drip or furrow irrigation. Infection occurs rapidly during periods of warm, wet weather. Fungicide sprays control the disease effectively.

Septoria leaf spots of tomato plants are caused by a fungus, which leads to circular water-soaked lesions that occur first on older leaves. These spots eventually turn brown with grey centres and die; if the infection is severe enough, the entire leaf will die. The fungus can survive in the debris from previous crops and/or weeds. Clean cultivation is important. Labelled fungicides can control the disease. The fungus causes the leaf mould disease of plants. Symptoms appear as light green patches on the upper surfaces of older leaves. Underneath the leaves in these areas, a purplish or olive-green patch of mould growth is visible. Infected leaves turn yellow and drop off the plant. Fungus is spread by wind currents. High humidity and warm temperatures encourage mould growth. The problem is especially severe in greenhouses, where adequate ventilation and air movement reduce disease severity by lowering moisture at the leaf surface. Fungicides are effective controls. The fungal pathogen causes the target spot of tomato. The initial foliar symptoms are pinpoint-sized, water-soaked spots on the upper leaf surface. The spots develop into small, necrotic lesions with light brown centres and dark margins. The primary strategy used to manage target spots on tomatoes is the regular application of fungicides.

Bacteria are also major plant pathogens. Bacterial spot is a plant disease caused by bacteria. First, dark brown water-soaked spots appear on the leaves; later, these spots become blackish, and eventually, the affected tissue drops out, leaving a hole in the leaf. Copper sprays provide some control. Good sanitation practices, including prompt plough-down of stubble and weed control, help prevent the disease.

Moulds are also a major cause of plant diseases. The characteristics of late blight of tomato plants caused by mould are that lesions on leaves appear as large water-soaked areas that eventually turn brown and papery. Fruit lesions are large irregular greenish-brown patches having a greasy rough appearance. Green to black irregular lesions are also present on the stems.

Tomato yellow leaf curl virus is a devastating virus causing tomato disease, denoted by plant stunting and pronounced chlorotic leaves that curl upward. Older leaves become leathery and brittle. The nodes and internodes are significantly reduced in size. The infected plants look pale and produce more lateral branches giving a bushy appearance. The infected plants remain stunted. Removal of plants with initial symptoms may slow the spread of the disease. Low-concentration sprays of a horticultural oil or canola oil will act as a whitefly repellent, reducing feeding and possibly transmission of the virus. Tomato mosaic virus is another viral disease.
Depending on the strain and age of plants when infected, plants are stunted with mosaic or fern leaf-like symptoms. Aphids often are

virus vectors, so an attempt to control the aphids is the first step. Eliminate weeds and remove infected plants from the field as soon as they are seen.

Mites are a common pest that attacks vegetable and fruit crops. Tomato plants attacked by mites often have a mottled or speckled dull appearance on the top leaf surfaces due to feeding injury. Leaves then turn yellow and drop. Large populations produce visible webbing that can completely cover the leaves. The use of overhead-sprinkler irrigation may provide short-term relief for mite infestations.

In-house and public tomato image datasets

To obtain a reliable CNN model, a large-scale dataset is essential. We used both in-house and public datasets of tomato disease leaf images in this work. The first dataset was acquired from the College of Water Conservancy and Civil Engineering greenhouse at South China Agricultural University. The tomato seedlings were inoculated with tomato yellow leaf curl virus (TYLCV) and tomato mosaic virus (TMV) at the Plant Protection Research Institute of Guangdong Academy of Agricultural Sciences, then transplanted to the conservatory (Polston *et al*., 1999). Tomato was raised after seven-day acclimatisation. During the period of growth, the Hoagland nutrient solution (Hoagland *et al*., 1950) (150ml) was manually sprayed every two days by using a sprinkling can to provide sufficient nutrients for the plants. Tomato images were collected as the dataset during the growth cycle by a digital camera (SONY DSC-HX400), which was set to adjust the focal length and aperture, auto white balance, and without flash. The image resolution was 5184×3888 pixels. To maintain the diversity of the data for the test of the generalisation ability of the algorithm, tomato images were captured in all the weather conditions.

The second dataset, PlantVillage data (Hughes *et al*., 2015), was an open-access dataset with more than 50,000 images of leaves, from which we extracted 18,160 images of 10 different tomato leaf classes. These images' size was 256×256. To complement the in-house dataset and enhance data diversity, we also created the third dataset as a supplement and named it the Internet dataset. We downloaded 295 images from multi-sources, such as the Arkansas Plant Diseases Database, the American Phytopathological Society database, the Bugwood image database, and several other academic sources. So, these images' sizes varied from 35×47 to 2156×2232. The sampled images of these three datasets are shown in Figure 1. The differences among the images from these three sources included the shooting environment, the growing season, and the growing location. The specific type of diseases and quantities of the images in these three datasets are shown

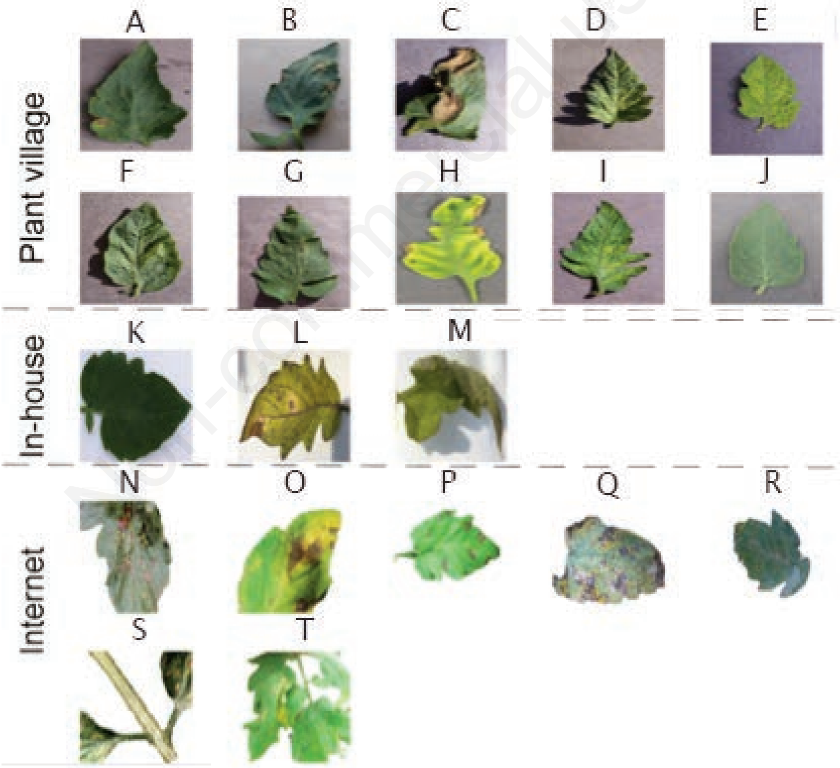

Figure 1. Sample images. A) PlantVillage data (PV) -Bacterial Spot; B) PV-Early Blight; C) PV-Late Blight; D) PV-Leaf Mould; E) PV-Septoria Leaf Spot; F) PV-Spider Mites two spotted Spider Mite; G) PV-Target Spot; H) PV-Tomato Yellow Leaf Curl Virus; I) PV-Tomato Mosaic Virus; J) PV-Healthy; K) OUR-Healthy; L) OUR-Tomato Mosaic Virus; M) OUR-Tomato Yellow Leaf Curl Virus; N) Internet data (INT)-Bacterial Spot (Clemson University - USDA Cooperative Extension Slide Series, Bugwood.org / CC BY 3.0); O) INT-Early Blight (Nancy Gregory, University of Delaware, Bugwood.org / CC BY 3.0); P) INT-Late Blight (Elizabeth Bush, Virginia Polytechnic Institute and State University, Bugwood.org / CC BY 3.0); Q) INT-Leaf Mould (Elizabeth Bush, Virginia Polytechnic Institute and State University, Bugwood.org / CC BY 3.0); R) INT-Septoria Leaf Spot (William M. Brown Jr., Bugwood.org / CC BY 3.0); S) INT-Spider Mites-Two Spotted Spider Mite (Clemson University - USDA Cooperative Extension Slide Series, Bugwood.org / CC BY 3.0); T) INT-Target Spot (Photograph by Dr. Ken Pernezny from University of Florida / CC BY-SA 4.0). PV, PlantVillage data released under CC BY-SA 3.0; INT, Internet data.

in Table 1. Since there are few other public large-scale tomato disease datasets, the Internet dataset was much smaller than the PlantVillage dataset. In addition, the image background of our in-house dataset and the PlantVillage dataset was uniform and noise-free. Nevertheless, most of the Internet dataset images were taken in the wild; their backgrounds were complicated and noisy. So, we manually segmented the background for the Internet dataset.

Image augmentation

Accurate and comprehensive annotation of the images for training and validation was crucial for developing an appropriate and reliable detecting model because deep learning models tend to fit the labelled samples for learning the features during the training. Thus, the ideal training images should be photographed from various angles, scales, and scopes, which was difficult to achieve in actual shooting situations because of the photograph restrictions. Furthermore, the lighting condition was another disturbance. Many factors interfered with the lighting condition, such as the weather, occlusion between leaves, shadows, and the disturbance of sand and dust. Only using these disturbing images as training data could lead to the over-fitting problem of deep learning models (Heisel *et al.*, 2017). Besides, both the in-house and Internet datasets were much smaller than the PlantVillage dataset. The augmented images were used to enrich the dataset at the experimental phase to expand the data amount and better generalisation.

Image preprocessing is a common technique for enhancing data (Bow, 2002). This technique involves removing low-frequency background noise, normalising the intensity of images of individual particles, removing reflections, and masking portions of images. In this work, we eliminated the interference of the shooting position by randomly rotating within 90-degree angles, flipping, and mirroring. We also shifted the brightness value to eliminate the problem of uneven illumination. In addition, cropping the original image randomly also augmented the original data. We used the ImageDataGenerator tool in Keras (Chollet, 2015) to augment the in-house and Internet datasets. ImageDataGenerator can generate batches of tensor image data with real-time data augmentation. The data will be looped over (in batches). The continuous iteration can keep producing the augmented image. After excluding the largest and smallest classes in the PlantVillage dataset, the median number of all classes was 952. The largest number of classes in the in-house and Internet datasets was 111. The difference between the two was about 9 times. To reduce the huge gap among the number of three datasets and avoid excessive redundancy of repeated images, we finally augmented 9 times the in-house dataset and Internet dataset. To expand the diversity of the training data, we merged the augmented in-house dataset and the Internet dataset as a new dataset. Then, we split this new dataset and the PlantVillage dataset into testing data (10%) and training data (90%), respectively. Finally, we combined the corresponding training and test data as the final input data. During the training process, 10% was split from the training data as a validation set to verify the model trained extent. The final input data are shown in Table 2.

Table 1. The consist of the dataset.

Diseases name	Pathogen	Quantities PlantVillage dataset	 In-house dataset	 Internet dataset
Tomato Bacterial Spot	Bacteria	2 127	0	34
Tomato Early Blight	Fungi	1 000	0	61
Tomato Healthy	-	1 591	82	0
Tomato Late Blight	Mould	1 909	0	63
Tomato Leaf Mould	Fungi	952	0	57
Tomato Septoria Leaf Spot	Fungi	1 771	0	51
Tomato Spider Mites/Two-spotted Spider	Mite	1 676	0	18
Tomato Target Spot	Fungi	1 404	0	11a
Tomato Mosaic Virus	Virus	373	110	0
Tomato Yellow Leaf Curl Virus	Virus	5 357	111	0
Total	-	18 160	303	295

Table 2. Input data.

Diseases name	Training set	Validation set	Test set
Tomato Bacterial Spot	1998	222	247
Tomato Early Blight	1305	144	161
Tomato Healthy	1953	216	242
Tomato Late Blight	2057	228	254
Tomato Leaf Mould	1233	136	153
Tomato Septoria Leaf Spot	1847	205	229
Tomato Spider Mites/Two-spotted Spider	1503	167	186
Tomato Target Spot	1226	136	152
Tomato Mosaic Virus	1193	132	148
Tomato Yellow Leaf Curl Virus	5 38	582	647
Total	19553	2 168	2 419

Deep convolution neural network and transfer learning

CNN has been developed explosively since its successful application in classifying the MNSIT (LeCun *et al*., 1998) dataset. The close connection and spatial information between layers in CNN make it particularly suitable for image processing and understanding. Deep learning is an end-to-end approach. Researchers (Dhingra *et al*., 2018; Durmuş *et al*., 2017) have proven that CNN can automatically extract rich, relevant features from images. Given a large amount of input image data and output labels, a CNN model automatically learns the features in the data. The learned features are effective because the data annotation is accurate. For a long time, increasing the hidden layer was a common strategy to improve the performance of the networks, and there are many excellent models which have been proposed based on that, such as AlexNet (Krizhevsky *et al*., 2012), VGG16 (Simonyan *et al*., 2014), Inception_v3 (Szegedy *et al*., 2015), ResNet50 (He *et al*., 2016) and DenseNet (Huang *et al*., 2017).

According to literature (Brahimi *et al*., 2018), six state-of-the-art architectures (AlexNet, DenseNet-169, Inception_v3, ResNet-50, SqueezeNet-1.1, and VGG13) were trained on the PlantVillage dataset. Inception_v3 gave the best accuracy for the deep training strategy. The inception in Inception_v3 is the most prominent characteristic. Its core idea is factorisation, replacing the big convolutional kernel with multiple smaller kernels. The inception mechanism not only reduces calculations but also improves feature extraction capabilities. However, when the network depth increases, the network accuracy becomes saturated or even decreases; this problem is called the degradation problem. ResNet was developed to solve this problem; it can achieve a deeper network without causing the gradient to disappear or the gradient explosion problem. VGG16 is one of the most utilised classical sequential networks. It was used to win the ILSVR (ImageNet) competition in 2014. Although its performance was surpassed by the later Inception and ResNet architectures, it still has research value because of its compact structure and easy-to-implement characteristics. So, we compared the performance of using the backbone of VGG16, Inception_v3, and ResNet50 on tomato disease images in this work.

Usually, the structure of a deep CNN contains the convolutional layer, fully connected layer, pooling layer, and other additional layers. The multi groups of convolutional layers are the core because they were used to gradually extract the advanced feature from the image. The fully connected layers act as classifiers throughout the CNN. The convolutional layer and pooling layer mapped the original data to the hidden layer feature space, the fully connected layer mapped the learned *distributed feature representation* to the sampled marked space. Because of the redundancy of the fully connected layer parameters (its parameters account for about 80% of the entire network parameters), some high-performance network models such as ResNet and DenseNet replaced the fully connected layer connected to the last pooling layer with global max pooling (GMP) (Lin *et al*., 2013). However, redundant parameters are not always worthless. Zhang *et al*. (2017) have shown that the fully connected layer can use as a *firewall* for model representation capability. Especially in the case where the source domain and the target domain have large differences, the fully connected layer can maintain a large model capacity to ensure the migration of the model representation capability.

Transfer learning is a machine learning technique in which a model trained on one task is re-purposed on a related task (Goodfellow *et al*., 2017). Fine-tuning is the most commonly used transfer learning technology. For example, VGG16, Inception_v3, and ResNet50 achieved spectacular results in the computer vision challenge ImageNet. Furthermore, these models allow us to fine-tune and transfer learning from a task with a large labeled dataset to a particular task such as disease classification. In the case of small training datasets, the pre-trained model achieved faster convergence and higher accuracy (Brahimi *et al*., 2017). In this work, we keep the feature extraction part of the pre-trained existing deep architectures but redesigned their classifier. We replaced the flattened layer with GMP, simplified the fully connected layers and their hidden nodes, and added batch normalisation layers. Simultaneously, we used the dropout layer before the output layer. The construction is shown in Figure 2.

Results

Deep convolution neural network can accurately classify tomato leaf disease images

The training proceeded on the training set; after that, the evaluation in each epoch was performed on the validation set, and the final evaluation was done on the testing set. The validation set is a technique used for minimising over-fitting and is a typical way to control training processing.

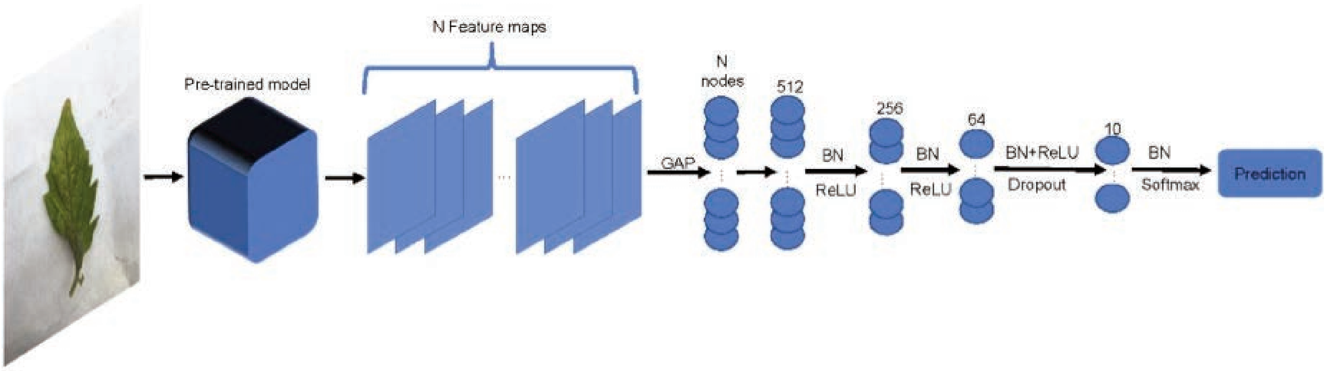

Figure 2. Network construction. We used VGG16 and Inception_v3 as the pre-trained model. N determined by the selected pre-trained model, for VGG16 is 512, for Inception_v3 is 2048, and for Resnet is 2048.

The network was trained using mini-batch stochastic gradient descent with a momentum factor. The number of samples per small batch was 32, and the momentum factor was set to a fixed value of 0.9. The initialisation of the weight affected the convergence speed of the network. In this work, the Glorot uniform initialiser (Glorot and Bengio, 2010) was used to initialise the weights of all network layers. The biases of all convolutional layers and fully connected layers were initialised to 0. The same learning rate was adopted for all layers in the network, and the initial learning rate was set to 0.001 and decayed 1e-6 for each training epoch.

All the processes of training and testing the tomato disease identification model described in this work were implemented on one machine, whose configuration parameters were Intel Core i9-7900X 3.3 GHz Processor, an Nvidia GeForce 2080Ti GPU, and 11GB memory. The model proposed in this work was trained on Tensorflow (Abadi *et al*., 2015) and Keras (Kotikalapudi, 2017) frameworks. During the training process, the augmented dataset was produced along with the increased number of iterations, which meant that the training dataset differed in each epoch. However, the model was saved once the validation loss was improved. The model convergence result is shown in Table 3. And Figure 3 illustrates the change of accuracy and loss in 300 epochs, demonstrating that those models efficiently learn the data while achieving a high accuracy rate.

From Table 3, all models converged before the 300th epoch, which means that the number of training epochs was sufficient. Resnet50 converged fastest among these three models; its loss arrived at 0.00281 at the 246th epoch. However, the test accuracy was inversely proportional to minimum validation loss. According to Figure 3, the training and validation accuracy curves were nearly flat after 200 epochs and close to 1. Observing the training process curve, we found that the validation loss converged to a small value and did not change in many epochs before the minimum validation loss was obtained. So, we set up the early stop strategy to avoid over-fitting. Once the validation loss does not change in 30 epochs, stop the training. As we can see from Table 3, an early stop strategy can significantly reduce model training time. Compared with the regular strategy, the early stop strategy ended the training about 100 epochs in advance and kept the test accuracy high. Finally, we choose the Inception_v3 model to package an application.

TomatoGuard application

For most of the similar research, they ended their work after completing the model training. To move forward a single step, applying the findings in practice, we deployed the proposed deep learning model into Android devices. Nowadays, the smartphone has become indispensable tool in people's life. Since 2007, the

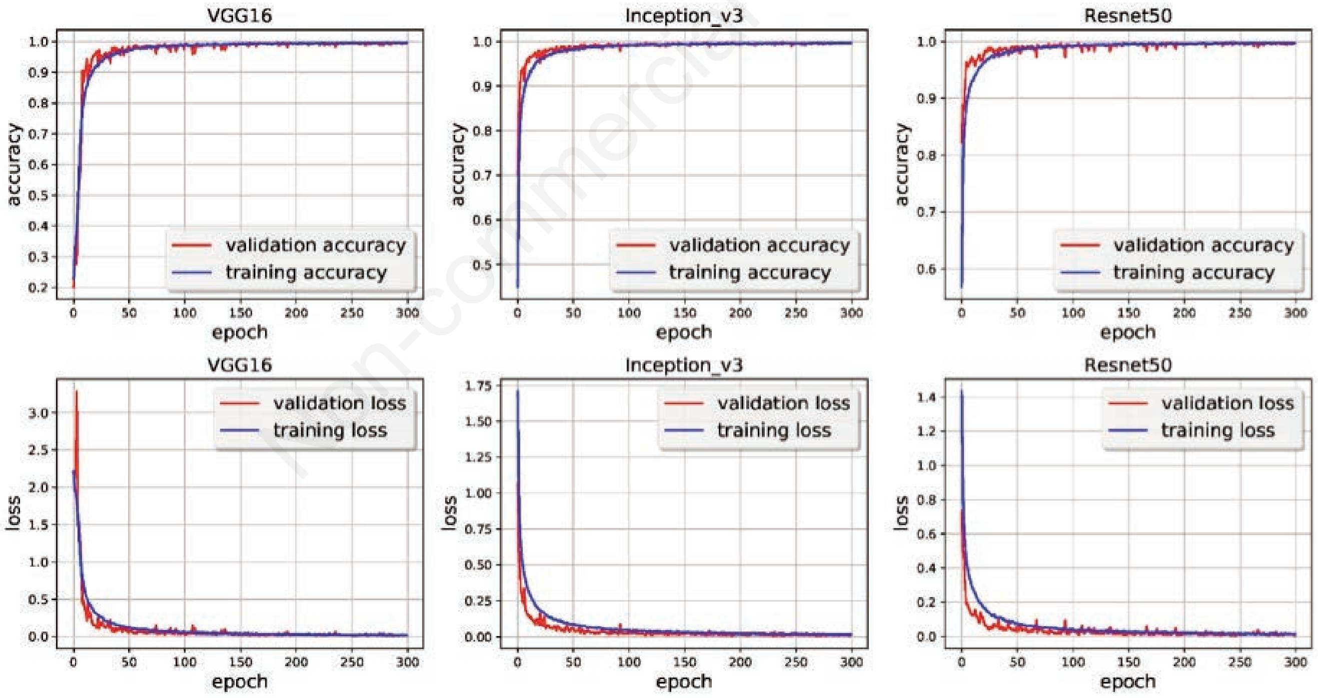

Figure 3. Model training process.

Table 3. Model convergence result.

Model	Regular			Add early stop		
	Min validation loss	Epoch	Test accuracy %	Min validation loss	Epoch	Test accuracy %
VGG16	0.00822	288	99.75	0.00856	162	99.62
Inception_v3	0.00491	273	99.42	0.00604	177	99.75
Resnet50	0.00281	246	98.92	0.00779	108	99.50

Android system has been widely used in smartphones and tablet computers. To implement our research, deploying the trained deep learning model in a smartphone is meaningful. We use Android Studio to deploy our model into the Android device. To optimise the model for smartphone applications, we compressed our trained model with Tensorflow lite and compared it. The model's performance before and after compression is shown in Table 4. All the test was run on Huawei Honor V8 equipped with the Android 8.0 system. Compared with regular model packaging applications, compressed model applications not only had smaller installation packages but also had faster calculation speeds. However, the test accuracy was slightly reduced. 12.6% reduction in installation package size in exchange for 8.3% accuracy reduction. Since both operation speeds were at the millisecond level, accuracy should be prioritised when hardware conditions permit. Our goal is to provide an immediate and effective tool for people who want to recognise tomato leaf diseases. So, the process of our application is inputting a tomato leaf image, computing it by the advanced trained model, and displaying the result of recognition. We developed an Android APP named TomatoGuard. It can identify 9 kinds of tomato leaf diseases from healthy tomato leaves. The APP workflow is shown in Figure 4. In the current version of TomatoGuard, we implemented captured images from the camera or loaded local pictures in the device. The input tomato leaf image needs to undergo image processing to divide all pixel values by 255 to meet the training model data requirements. Image processing steps are done automatically. The identification of tomato disease types and confidence levels are quickly displayed above the image. At present, there is only one similar APP in Google Play called *Plantix*. It claimed that it could detect more than 200 diseases covered from more than 40 crops, including 10 kinds of tomato leaf images discussed in this work. However, since some of the test set in this work was augmented from some of the training set, the test set was unsuitable for testing Plantix. So, we collected 127 more new images, including 10 classes mentioned above from multi-sources for comparing with Plantix strictly. The test result is shown in Figure 5.

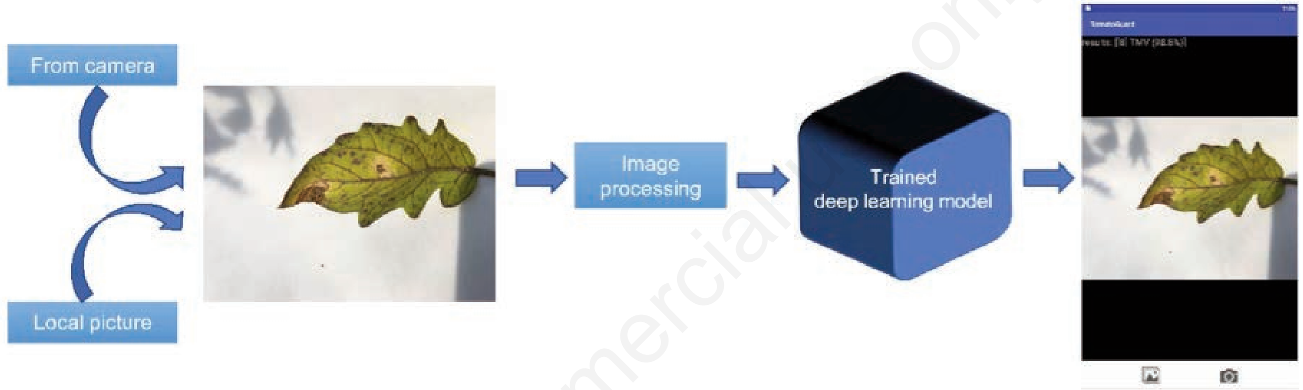

Figure 4. The workflow of TomatoGuard.

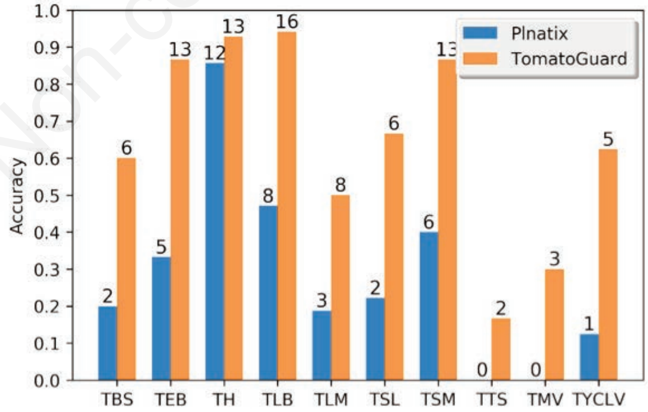

Figure 5. Test result. The number above the column is the quantity of the image. TBS, tomato bacterial spot; TEB, tomato early blight; TH, tomato healty; TLB, tomato late blight; TLM, tomato leaf mould; TSL, tomato septoria leaf spot; TSM, tomato spider mites/two-spotted spider mites; TTS, tomato target spot; TMV, tomato mosaic virus; TYLCV, tomato yellow leaf curl virus.

Table 4. Android application test results.

Model	Installation package size	Per image average calculate speed	Test accuracy %
Regular model	103M	360ms	67.5
Compressed model	90M	220ms	61.9

TomatoGuard performed better than Plantix. Additionally, for low-resolution images, even if users have selected the objective crop in advance, Plantix asked users to repeatedly choose the kind of crop during testing because of its compatibility and versatility.

From the perspective of early detection of diseases, accurate identification is the basic premise, and rapid feedback and low cost are the core conditions for a useful implementation to be extended to ordinary farmers. As shown above, TomatoGuard basically meets the requirements for realising early tomato disease identification. However, the current version of TomatoGuard stays at the technology readiness level (TRL) 5, and the application needs to optimise the interaction logic and supplement more functions before it is put on the market.

Feature visualisation

The symptoms of tomato diseases are various. Some disease characteristics are spots on the leaf or/and the edge of the leaf. Some diseases infect the whole leaf, it which will change the leaf colour, shape, and texture, such as TYLCV. These diseases' characterisation is a global feature. These tomato disease diagnoses depend on the entire leaf. Therefore, it is unsuitable to localise these disease parts on the leaf.

We visualised the model output layer to analyse how the neural network works with an image. First, we generated 10 arbitrary images and updated them until we maximised each node activation of the network's last layer. These images represented the feature of each class exacted from this model and allowed us to understand what sort of input patterns activate a particular filter, as shown in Figure 6. Distinctly, most of these features do not conform to any observation in the human sense. This is because the CNN classifies images by decomposing the visual input space into a hierarchical-modular network of convolution filters mapping the probabilities between certain combinations of these filters and a set of arbitrary labels. But some class feature maps made sense, such as tomato bacterial spot (TBS) feature map containing multiple prominent bright spots, similar to TBS pathological symptoms.

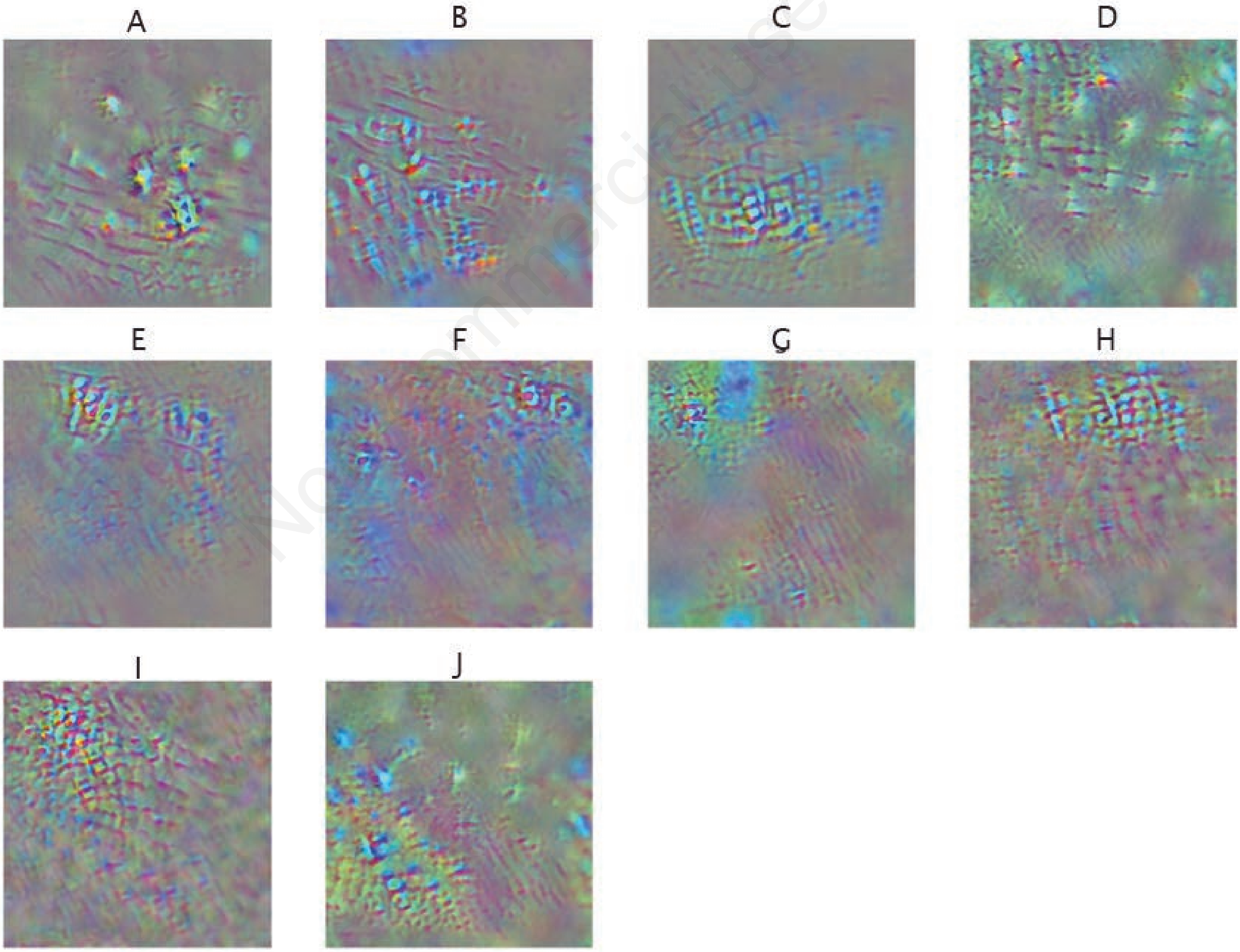

Figure 6. Feature maps. A) Tomato bacterial spot; B) Tomato early blight; C) Tomato late blight; D) Tomato leaf mould; E) Tomato septoria leaf spot; F) Tomato spider mites/two-spotted spider mites; G) Tomato target spot; H) Tomato yellow leaf curl virus; I) Tomato mosaic virus; J) Tomato helathy.

Discussion

Because of the complexity of the patterns shown in each class, especially in terms of infection status and background, the model tends to be confused with several classes. Figure 7 presents a confusion matrix of the Inception_v3 test results.

Based on the confusion matrix, we can evaluate the classifier's performance and determine what classes were more highlighted by the neurons in the network. Furthermore, it helped us analyse further procedures to avoid inter-class confusion. For instance, 2 tomato target spot (TTS) images were being incorrectly classified into tomato spider mites (TSM). The TSM class precision (98.9%) was shown to be the lowest. These two classes have shown relatively strongly missed up. This was related to their disease symptoms. TSM is not a virus disease, spider mites cause it. Mostly mites live on the underside of leaves and feed by piercing leaf tissue and sucking up plant fluids. Feeding marks show up as light dots on the leaves. As feeding continues, the leaves turn yellow. The

	TBS	TEB	TH	TLB	TLM	TMV	TSL	TSM	TTS	TYLCV	
TBS	247 10.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
TEB	0 0.0%	159 6.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
TH	0 0.0%	0 0.0%	242 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
TLB	0 0.0%	0 0.0%	0 0.0%	254 10.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
TLM	0 0.0%	0 0.0%	0 0.0%	0 0.0%	153 6.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
TMV	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	147 6.1%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	99.3% 0.7%
TSL	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	229 9.5%	0 0.0%	0 0.0%	0 0.0%	99.6% 0.4%
TSM	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	186 7.7%	2 0.1%	0 0.0%	98.9% 1.1%
TTS	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	150 6.2%	0 0.0%	99.3% 0.7%
TYLCV	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	646 26.7%	99.8% 0.2%
	100% 0.0%	98.8% 1.2%	100% 0.0%	100% 0.0%	100% 0.0%	99.3% 0.7%	100% 0.0%	100% 0.0%	98.7% 1.3%	99.8% 0.2%	99.8% 0.2%

Figure 7. Confusion matrix. The rows (Output Class) correspond to the predicted class, and the columns (Target Class) correspond to the true class. Diagonal cells are the correctly classified observations. Off-diagonal cells are the incorrectly classified observations. Both the number of observations and the percentage of the total number of observations are shown in each cell. The rightmost column of the plot shows the percentages of all the examples predicted to belong to each class that are correctly and incorrectly classified. These metrics are often called the precision (or positive predictive value) and false discovery rate, respectively. The row at the bottom of the plot shows the percentages of all the examples belonging to each class that are correctly and incorrectly classified. These metrics are often called the recall (or true positive rate) and false negative rate. Finally, the cell in the bottom right of the plot shows the overall accuracy. TBS, tomato bacterial spot; TEB, tomato early blight; TH, tomato healty; TLB, tomato late blight; TLM, tomato leaf mould; TSL, tomato septoria leaf spot; TSM, tomato spider mites/two-spotted spider mites; TTS, tomato target spot; TMV, tomato mosaic virus; TYLCV, tomato yellow leaf curl virus.

TTS disease first appears as small, necrotic lesions with light-brown centers and dark margins on tomato leaves. In brief, these diseases have common symptoms: they all have spots (marks) on the leaf and change the colour to yellow. Moreover, two tomato early blight (TEB) images were misidentified as TTS (1) and tomato septoria leaf spot (TSL) (1). Comparing TEB, TTS, and TSL, they all have similar disease spots. TEB causes small dark spots to form on older foliage near the ground initially. Leaf spots are round and brown and can grow up to a half-inch in diameter. Larger spots have a target like concentric rings, and tissue around spots often turns yellow (Gleason and Edmunds, 2005).

Similarly, spots caused by TSL are circular with dark brown margins and tan to gray centers with small black fruiting structures. Characteristically, there are many spots per leaf. If leaf lesions are numerous, the leaves turn slightly yellow, brown, and wither. When these diseases' unique characteristics are not obvious, there is the possibility of being confused. There are also 1 tomato mosaic virus (TMV) image classified as TYLCV and 1 TYLCV image classified as TMV. Considering the dense tomato growth environment of the in-house dataset, tomato plants may be cross-infected, which means there are probably two disease features shown on one leaf. Overfitting is the most prone problem in deep neural networks. Adding the dropout and batch normalisation layers (BN) to the model is a very effective way to solve the problem of overfit-

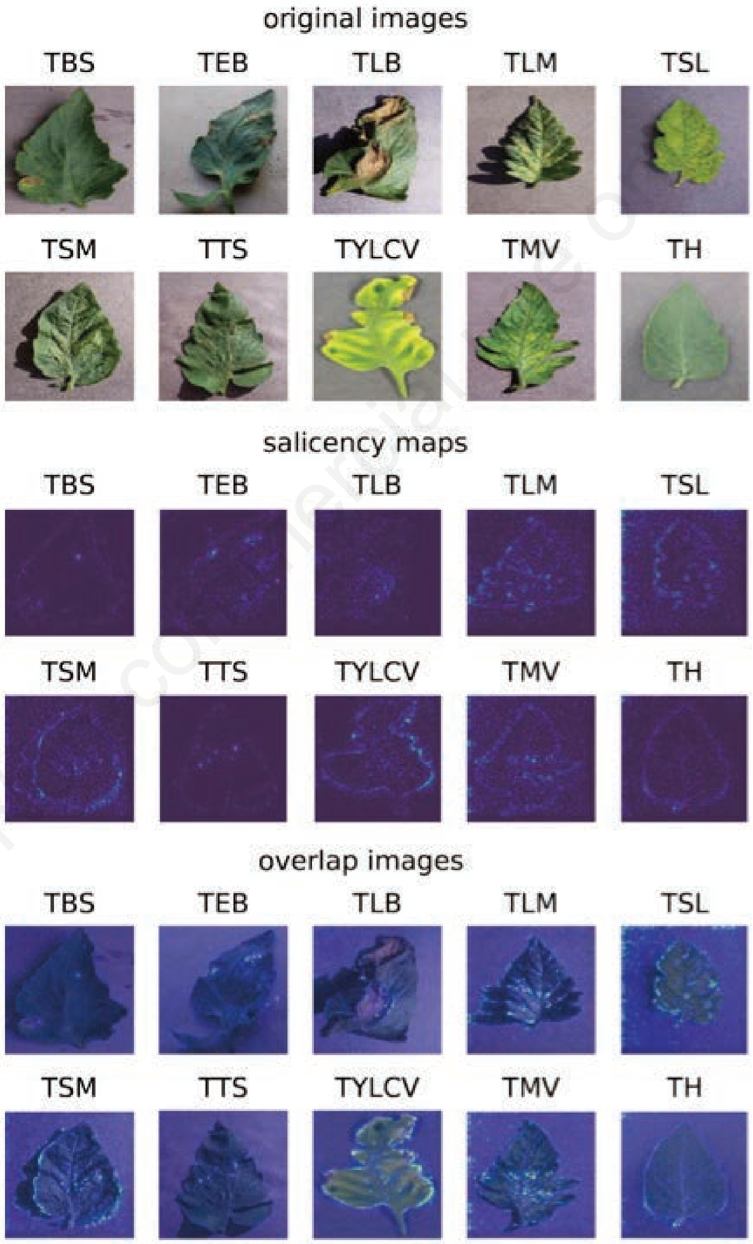

Figure 8. Saliency maps. The overlap images show the trained model interested area of diseased leaves. The original images were sampled from the PlantVillage data, released under the CC BY-SA 3.0. TBS, tomato bacterial spot; TEB, tomato early blight; TH, tomato healty; TLB, tomato late blight; TLM, tomato leaf mould; TSL, tomato septoria leaf spot; TSM, tomato spider mites/two-spotted spider mites; TTS, tomato target spot; TMV, tomato mosaic virus; TYLCV, tomato yellow leaf curl virus.

Table 5. Comparison experiment.

Method	Strategy							
Segmentation	×		×	×			×	
BN+Dropout	×	×		×		×		
GMP	×	×	×		×			
Test accuracy %	38.1	31.4	55.5	44.1	45.3	32.7	56.5	53.2
Test accuracy (Balanced data) %	31.2	29.3	51.4	44.3	34.3	36.8	39.4	31.1

BN, batch normalisation; GMP, global max pooling.

ting. The idea of the dropout layer can be described very simply: randomly drop units (along with their connections) from the neural network during training. The batch normalisation layer is used to normalise the activation of the previous layer at each batch, specifically applying a transformation to maintain the mean activation close to 0 and the activation standard deviation close to 1. Usually, batch normalisation and dropout layers can be used simultaneously. To verify which method was useful, we used different strategies to train VGG16 on 3 kinds of tomato disease [tomato healthy (TH), TMV, TYCLV] in PlantVillage data and tested them on our in-house data. Since the number of TYCLV classes in the PlantVillage data was much larger than the other two classes, we also did a comparison experiment with the balanced data (each class included 300 training images). The comparison experiment results are shown in Table 5. According to the experiment results, although the test accuracy was not high, the batch normalisation layer and dropout layer showed powerful improvement in the training effect. It should be pointed out that networks that use GMP instead of the fully connected layer have better predictive performance. In this experiment, the balanced data did not bring higher accuracy, even reduced accuracy. To explore whether the knowledge of model learning is consistent with the knowledge of pathology, we used a saliency map (Gleason and Edmunds, 2005) to figure out what corresponding pixel in a disease image our model used. A saliency map is an analytical method that allows the estimation of the importance of each pixel. The saliency maps are shown in Figure 8. From these images, it was easy to conclude that the saliency map can localise our model interest regions in the input tomato image with good precision. TEB, TLB, TLM, TSL, and TTS were good examples to illustrate that for each different tomato disease which pixels our model precisely focused on. The saliency areas of these diseases were the disease spots. For TSM and TYLCV, our model focused on the spider mites and the leaf edge, respectively. All these saliency areas were consistent with disease characteristics.

Conclusions

This work presents a deep-learning-based classifier for nine tomato diseases and healthy tomato leaf recognition, which achieved 99% test accuracy. We also developed a tool named TomatoGuard on Android devices to help people recognise tomato diseases. The experiment results showed that TomatoGuard dramatically overcame the state-of-the-art of this field as it recorded a higher test accuracy resulting in higher performance than the APP Plantix. We expect this tool to significantly contribute to the crop protection research area. Although TomatoGuard stays at the TRL5 level, to make this tool more applicable, future studies will need to detect the diseased crop planted on site, and the application needs to optimise interaction logic and supplement more functions.

References

Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jia Y., Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., Mane D., Monga R., Moore S., Murray D., Chris O., Schuster M., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viegas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., Zheng X. 2015. TensorFlow: large-scale machine learning on heterogeneous systems. arXiv 1603:04467.

Bow S. T. 2002. Pattern recognition and image preprocessing. Boca Raton, FL: CRC press.

Brahimi M., Arsenovic M., Laraba S., Sladojevic S., Boukhalfa K., Moussaoui A. 2018. Deep learning for plant diseases: detection and saliency map visualisation. In: Human and machine learning. Springer. pp. 93-117.

Brahimi M., Boukhalfa K., Moussaoui, A. 2017. Deep learning for tomato diseases: classification and symptoms visualization. Appl. Artif. Intell. 31:299-315.

Chollet F. 2015. Keras, GitHub. Available from: https://github.com/fchollet/keras

Chowdhury M.E., Rahman T., Khandakar A., Ibtehaz N., Khan A.U., Khan M.S. Al-Emadi N., Reaz M.B.I, Islam M.T, Ali S.H.M. 2021. Tomato leaf diseases detection using deep learning technique. In: Technology in Agriculture. Intech Open.

Chung C.L., Huang K.J., Chen SY., Lai M.H., Chen Y.C., Kuo Y.F. 2016. Detecting Bakanae disease in rice seedlings by machine vision. Comput. Electron. Agric. 121:404-11.

Dhingra G., Kumar V., Joshi H.D. 2018. Study of digital image processing techniques for leaf disease detection and classification. Multimed. Tools. Appl. 77:19951-20000.

Durmuş H., Güneş E.O., Kırcı M. 2017. Disease detection on the leaves of the tomato plants by using deep learning. In: 2017 6th International Conference on Agro-Geoinformatics. Fairfax, VA. pp. 1-5. IEEE.

Fuentes A., Yoon S., Kim S.C., Park D.S. 2017. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors. 17:2022.

Fukushima K., Miyake S., Ito T. 1983. Neocognitron: a neural network model for a mechanism of visual pattern recognition. IEEE. Trans. Syst. Man. Cybernet. 13:826-34.

Gleason M.L., Edmunds B.A. 2005. Tomato diseases and disorders. Agr. Sci. 7.

Glorot X., Bengio Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR 9:249-56.

Goodfellow I., Bengio Y., Courville A. 2017. Deep learning (adaptive computation and machine learning series). Cambridge, Massachusetts: The MIT Press. pp. 321-59.

Hanssen I.M., Lapidot M., Thomma B.P. 2010. Emerging viral diseases of tomato crops. Mol. Plant. Microbe. Interact. 23:539–48.

He K., Zhang X., Ren S., Sun J. 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Las Vegas, NV. pp. 770-8.

Heisel S., Kovačević T., Briesen H., Schembecker G., Wohlgemuth K. 2017. Variable selection and training set design for particle classification using a linear and a non-linear classifier. Chem. Eng. Sci. 173:131-44.

Hoagland D.R., Arnon D.I. 1950. The water-culture method for growing plants without soil. Circular. 347:32.

Hughes D., Salathé M. 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060.

Huang G., Liu Z., Van Der Maaten L., Weinberger K. Q. 2017. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Honolulu, HI. pp. 4700-8.

Kawasaki Y., Uga H., Kagiwada S., Iyatomi H. 2015. Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. In: International symposium on visual computing. Springer. pp. 638-645.

Kezhu T., Yuhua C., Weixian S., Xiaoda C. 2014. Identification of diseases for soybean seeds by computer vision applying BP neural network. Int. J. Agr. Biol. Eng. 7:43-50.

Kotikalapudi R. 2017. Keras Visualization Toolkit. Available from: https://github.com/raghakot/keras-vis.

Krizhevsky A., Sutskever I., Hinton G.E. 2012. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems 25. Available from: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf .

LeCun Y., Bottou L., Bengio Y., Haffner P. 1998. Gradient-based learning applied to document recognition. Proceedings. IEEE. 86:2278-324.

LeCun Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel, L.D. 1989. Backpropagation applied to handwritten zip code recognition. Neural. Comput. 1:541-51.

Lin M., Chen Q., Yan S. 2013. Network in network. arXiv preprint arXiv:1312.4400.

Liu B., Zhang Y., He D. Li Y. 2017. Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry.10:11.

Liu N., Han J., Zhang D., Wen S., Liu T. 2015. Predicting eye fixations using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 362-70.

Lu Y., Yi S., Zeng N., Liu Y., Zhang Y. 2017. Identification of rice diseases using deep convolutional neural networks. Neurocomputing. 267:378-84.

Masood S.Z., Shu G., Dehghan A., Ortiz, E.G. 2017. License plate detection and recognition using deeply learned convolutional neural networks. arXiv preprint arXiv:1703.07330.

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., Hassanien A.E., Hefny H. 2015. Identifying two of tomatoes leaf viruses using support vector machine. In: Information systems design and intelligent applications. Comput. Sci. 771-82.

Parkhi O.M., Vedaldi A., Zisserman A. 2015. Deep face recognition. BMVC 1:41.1-41.12.

Polston J.E., McGovern R.J., Brown, L.G. 1999. Introduction of tomato yellow leaf curl virus in Florida and implications for the spread of this and other geminiviruses of tomato. Plant. Dis. 83:984-8.

Prasad S., Singh P.P. 2017a. Medicinal plant leaf information extraction using deep features. In: TENCON 2017-2017 IEEE region 10 conference. IEEE. pp. 2722-6.

Prasad S., Singh P.P. 2017b. Vision system for medicinal plant leaf acquisition and analysis. In: Applications of cognitive computing systems and IBM Watson. Springer. pp. 37-45.

Riley M.B., Williamson M.R., Maloy O. 2002. Plant disease diagnosis. The plant health instructor. DOI: 10.1094/PHI-I-2002-1021-01

Romualdo L.M., Luz P.H.C., Devechio, F.F.S., Marin M.A., Zúñiga A.M.G., Bruno O.M., Herling V.R. 2014. Use of artificial vision techniques for diagnostic of nitrogen nutritional status in maize plants. Comput. Electron. Agr. 104:63-70.

Sabanci K., Kayabasi A., Toktas A. 2017. Computer vision-based method for classification of wheat grains using artificial neural network. J. Sci. Food. Agr. 97:2588-93.

Shrivastava S., Singh S.K., Hooda D.S. 2017. Soybean plant foliar disease detection using image retrieval approaches. Multimed. Tools. Appl. 76:26647-74.

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

Sladojevic S., Arsenovic M., Anderla A., Culibrk D., Stefanovic D. 2016. Deep neural networks based recognition of plant diseases by leaf image classification. Comput. Intel. Neuroscience. 2016.

Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Rabinovich, A. 2015. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Boston, MA. pp. 1-9.

Tamaazousti Y., Le Borgne H., Hudelot C., Tamaazousti M. 2020. Learning more universal representations for transfer-learning. IEEE. Trans. Pattern. Anal. Mach. Intel. 42:2212-24.

Tomè D., Monti F., Baroffio L., Bondi L., Tagliasacchi M., Tubaro S. 2016. Deep convolutional neural networks for pedestrian detection. Signal processing: image communication. arXiv:1510.03698. 47:482-89.

Wolfert S., Ge L., Verdouw C., Bogaardt M.J. 2017. Big data in smart farming–a review. Agric. Syst. 153:69–80.

Zhang C.L., Luo J.H., Wei X.S., Wu J. 2017. In defense of fully connected layers in visual representation transfer. In: Pacific Rim Conference on Multimedia. Comput. Sci. 807-17.