Chapter 6
Deep Learning for Plant Diseases: Detection and Saliency Map Visualisation

Mohammed Brahimi, Marko Arsenovic, Sohaib Laraba, Srdjan Sladojevic, Kamel Boukhalfa and Abdelouhab Moussaoui

Abstract Recently, many researchers have been inspired by the success of deep learning in computer vision to improve the performance of detection systems for plant diseases. Unfortunately, most of these studies did not leverage recent deep architectures and were based essentially on AlexNet, GoogleNet or similar architectures. Moreover, the research did not take advantage of deep learning visualisation methods which qualifies these deep classifiers as black boxes as they are not transparent. In this chapter, we have tested multiple state-of-the-art Convolutional Neural Network (CNN) architectures using three learning strategies on a public dataset for plant diseases classification. These new architectures outperform the state-of-the-art results of plant diseases classification with an accuracy reaching **99.76%**. Furthermore, we have proposed the use of saliency maps as a visualisation method

M. Brahimi (✉) · K. Boukhalfa
Department of Computer Science, USTHB University, Algiers, Algeria
e-mail: m_brahimi@esi.dz

K. Boukhalfa
e-mail: kboukhalfa@usthb.dz

M. Brahimi
Department of Computer Science, Mohamed El Bachir El Ibrahimi University, Bordj Bou Arreridj, Algeria

M. Arsenovic · S. Sladojevic
Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia
e-mail: arsenovic@uns.ac.rs

S. Sladojevic
e-mail: sladojevic@uns.ac.rs

S. Laraba
TCTS Lab, Numediart Institute, University of Mons, Mons, Belgium
e-mail: sohaib.laraba@umons.ac.be

A. Moussaoui
Department of Computer Science, Setif 1 University, Setif, Algeria
e-mail: moussaoui.abdel@gmail.com

© Springer International Publishing AG, part of Springer Nature 2018

J. Zhou and F. Chen (eds.), *Human and Machine Learning*, Human–Computer Interaction Series, https://doi.org/10.1007/978-3-319-90403-0_6

to understand and interpret the CNN classification mechanism. This visualisation method increases the transparency of deep learning models and gives more insight into the symptoms of plant diseases.

6.1 Introduction

Plant diseases can cause great damages to agriculture crops by significantly decreasing production [12]. Early blight is a typical example of disease that can severely decrease production [4]. Similarly, in a humid climate, late blight is another very destructive disease that affects the plant leaves, stems, and fruits [4]. Protecting plants from diseases is vital to guarantee the quality and quantity of crops [5]. A successful protection strategy should start with an early detection of the disease in order to choose the appropriate treatment at the right time to prevent it from spreading [2]. Usually, this detection is achieved by experts having an academic knowledge reinforced by practical experience on symptoms and causes of diseases [4]. Furthermore, these experts must monitor plants consistently to avoid disease spreading. This continuous monitoring represents a difficult and time-consuming task for humans, which makes the automation of the plant diseases detection and identification essential to protect plants [5]. Several studies [1, 2, 7, 26] have been proposed to detect and classify plant diseases using image processing and machine learning. These approaches try to build disease classifiers using images taken from the crops. These classifiers are based on hand-crafted features designed by experts to extract relevant information for image classification. For this reason, these classifiers suffer from the lack of automation because of the dependency on hand-crafted features [22]. Moreover, the classifier must be trained using images labelled by experts. Collecting these labelled images is very expensive because it is done manually. This difficulty of data collection has forced the previous studies to use small datasets to train and test classifiers [1, 2, 7, 26]. The use of small labelled datasets is a limiting factor in machine learning, and it can lead to overfitting. In the last few years, Deep Learning (DL) has been adopted by the computer vision community, thanks to its results that outperform the state-of-the-art in many domains. The main advantage of DL in computer vision is the direct exploitation of image without any hand-crafted features. DL classifiers are end-to-end systems that form features in a fully automated way without any intervention by human experts. In plant diseases protection, many works have proposed the use of DL to detect and classify diseases. Notably, in [15] more than 54,306 images of diseased and healthy plant leaves are collected which makes the training of DL classifier possible. This new trend produced more accurate classifiers compared to traditional machine learning approaches [5, 8, 9, 18, 24, 25, 27, 34]. Despite these good results, DL research in plant diseases remains immature and requires more attention to produce practical systems. For example, many new successful deep architectures are not tested in the context of plant diseases. Moreover, DL classifiers suffer from a lack of interpretability and transparency. These accurate classifiers are often considered as black boxes that give good results but without any explanation or details about the classification mechanism. High accuracy is not sufficient for plant disease classification. Users also need to be informed how the detection is achieved and which symptoms are present in the plant. This knowledge is very important from a practical viewpoint. For example, inexperienced farmers can gain intuition about disease and symptoms used by the classifier. Similarly, agriculture experts and experienced farmers can evaluate the classifier decision by showing its classification mechanism. Also, these experts can exploit the transparency of the classifier to discover new symptoms or to localise known symptoms that are difficult to see with the human eye. In this chapter, we will compare previous works based on DL to detect and identify diseases. Moreover, we evaluate the state-of-the-art deep architectures based on the dataset proposed in [15]. Furthermore, we investigate visualisation methods applied on deep models to increase the transparency of deep classifiers. This study presents two main contributions in plant disease classification:

- Comparison between state-of-the-art CNN architectures performance in plant diseases protection: this comparison helps researchers to choose the best deep architecture for building a practical system for plant diseases protection.
- Visualisation of symptoms used by deep classifiers: visualisation methods allow the localisation of the infected region on the plant and help the users by giving them information about the disease. Also, this biological information is extracted without the intervention of agriculture experts. In this study, we propose the saliency map as a visualisation method based on a derivative of the deep network output with respect to the image.

6.2 Related Work

Plant diseases classification can be a very complex task as it relies mainly on experts know-how. Developing a reliable system that is applicable for a large number of classes is a very challenging task. Up to now, most of the approaches for automatic plant diseases classification depended on machine learning algorithms and basic feature engineering. These approaches are usually concentrated on certain environments and are suited for a smaller number of classes, where some small changes in the system can result in a drastic fall in accuracy. In recent years, Convolutional Neural Networks (CNN) have shown great results in many image classification tasks which have given researchers the opportunity to improve classification accuracy in many fields including agriculture and plant diseases classification.

Kawasaki et al. [18] proposed the use of deep CNN to distinguish healthy cucumbers from the infected ones by using images of leaves. In this study, they used CNN to diagnose two harmful viral infections: MYSV (melon yellow spot virus) and ZYMV (zucchini yellow mosaic virus). The used dataset in this work consists of 800 images of cucumbers leaves (300 with MYSV, 200 with ZYMV and 300 non-diseased). Rotation transformations on images were used to enlarge the dataset. For this binary classification task, authors proposed CNN architecture which consists of three convolutional layers, pooling layers, and local contrast normalisation layers. The activation function used in this network is the Rectified Linear Unit (ReLU) function. The achieved accuracy of this study is 94.9% under a 4-fold cross-validation strategy.

Sladojevic et al. [34] applied deep CNN for plant diseases classification. They collected a dataset from publicly available images on the internet. This dataset contains 13 classes of plant diseases, one class of healthy leaves and one class of background images extracted from Stanford background dataset [11]. The addition of this background class is to train the classifier to distinguish the plants leaves from the background images, which is emphasised as the limitation of [25]. The obtained dataset contains 4483 original images of different sizes, qualities, and backgrounds. To increase the size of this dataset, images were pre-processed and augmented to reduce overfitting in the training stage. For the augmentation stage, affine and perspective transformation were used in addition to image rotations. Using these augmentation transformations, the produced dataset consists of 30880 images for training and 2589 images for validation. Authors proposed the transfer learning using CaffeNet architecture [17]. CaffeNet is a modified version of AlexNet architecture that switches the order of pooling and the normalisation layers. This CNN was trained with and without fine-tuning by experimental changing of hidden layers parameters and hyperparameters. Visualisation of the features in the trained classification model intuitively helped in understanding the network which aided the fine-tuning process. The overall accuracy of the best architecture was 96.3% with fine-tuning and 95.8% without fine-tuning.

Mohanty et al. [25] used the public dataset named PlantVillage [15] which consists of 38 labelled classes including 26 diseases of 14 crop species. Authors used three versions of the dataset. The first version contains colour images, the second one contains grey-scaled images, and the third one contains images of segmented leaves to assess the influence of the background information on classification. Different training-test distributions were used to measure the performance of the CNN; 80–20, 60–40, 50–50, 40–60 and 20–80%. Two standards architectures were used for classification, AlexNet and GoogLeNet. They used two training strategies for training the CNN; training from scratch and transfer learning. They used 60 experimental configurations (2 CNN architectures ×3 versions of the dataset ×2 types of training ×5 training-test distributions) to evaluate the accuracy of deep CNN for the plant diseases classification task. From all the configurations, the highest accuracy is 99.34% which was achieved by the transfer learning of GoogleNet on the colour images using 80–20 dataset distribution. However, a couple of limitations were underlined in this study. Firstly, the majority of images are captured in a controlled environment using a simple background. Secondly, the number of images is not sufficient to train a classifier that is able to generalise to images taken in an uncontrolled environment. For instance, authors achieve an evaluation of the trained model using images taken from different conditions which shows that the accuracy decreases significantly to 31%.

Nachtigall et al. [27] proposed a CNN for automatic detection and classification of nutritional deficiencies and damages on apple trees from images of leaves. To build a dataset, they collected healthy leaves, two classes of damage caused by nutritional imbalances, two classes of diseases and one class of damage caused by the herbicide. To ensure the quality of labelling, chemical analysis was conducted for symptoms caused by nutritional imbalances and herbicide damage. This well-balanced dataset of 1450 images contains 290 images in each class (15 for testing, 193 for training and 83 for validation). AlexNet was used as a CNN architecture for building a classifier. They compared shallow methods against deep CNN. For the shallow method, Multilayer Perceptron (MLP) was chosen. Deep CNN was compared with seven volunteer experts where the final diagnoses were chosen by majority vote. The accuracy of the CNN was 97.3%, human experts had an accuracy of 96% where much less accuracy was achieved by MLP at 77.3%.

Fujita et al. [9] proposed a classifier for cucumber diseases using CNN. They used two datasets for training and validation. These datasets contain seven different types of diseases in addition to the healthy class. The first dataset consists of 7320 centred images of leaves captured under good conditions. The second dataset consists of 7520 images captured under good and bad conditions. To increase the size of this dataset, many crops from each image are used in addition to the rotated and the mirrored images. The proposed network is composed of four convolutional layers alternated with max-pooling layers and local response normalisation functions having parameters from AlexNet architecture [21]. Finally, the accuracy of the proposed system was 82.3% under a 4-fold cross validation scenario test.

Brahimi et al. [5] applied CNN for classifying tomato diseases based on images of leaves. In this study, the dataset consists of 14828 images of tomato leaves extracted from PlantVillage public dataset and divided into nine classes of diseases. For developing a classifier, the standard architectures AlexNet and GoogLeNet were trained from scratch or using transfer learning and fine-tuning. CNN models trained with fine-tuning showed a better accuracy than models trained from scratch. For GoogleNet, fine-tuning improves the accuracy from 97.71 to 99.18% and similarly for AlexNet the fine-tuning increases the accuracy from 97.35 to 98.66%. Authors have also compared the accuracy of the CNN with shallow models; SVM and Random Forest. CNN models have a better accuracy than shallow models, 94.53% for SVM and 95.46% for Random Forest. Finally, authors have proposed the use of occlusion experiments for localising and visualising the diseases regions and symptoms which can help users by giving them better insight to the diseases.

DeChant et al. [8] proposed to use the DL approach for the classification of northern leaf blight lesions on images of maize plants. 1028 images of infected leaves and 768 images of non-infected leaves were gathered on the field. From the total number of images, 70% were used for training, 15% for validation and 15% for testing. The proposed classification method in this chapter differs from the other studies presented in this chapter. In this study, instead of using only one end-to-end network in classification, the authors applied three training stages. In the first stage, several CNN models were trained to detect the presence of lesions in small parts of the images. These CNN models were used in the second stage to produce a heat map indicating the probability of infection for every image. In the last stage, the produced

Table 6.1 Comparison between deep learning studies for plant diseases classification

Paper	Year	Nbr. of classes	Nbr. of images	Image pre-processing and augmentation	CNN architecture	Transfer learning	Accuracy %
[18]	2015	3	800	Centred crop Resizing Rotation	Customised	No	94.90
[34]	2016	15	4483	Rotation Affine transformation Perspective transformation	CaffeNet	Yes	96.30
[25]	2016	38	54306	Resizing Segmentation Grey-scaling	AlexNet GoogLeNet	Yes	99.34
[27]	2016	5	1450	Resizing	AlexNet	Yes	97.30
[9]	2016	7	14840	Centred crop Resizing Rotation Shifting and Mirroring	Customised	No	82.30
[5]	2017	9	14828	Resizing	AlexNet GoogLeNet	Yes	99.18
[8]	2017	2	1796	Segmentation Rotation	Customised	No	96.70
[24]	2017	10	500	Resize Grey-scaling	AlexNet	Yes	95.48

heat map was used to classify the images. The proposed system achieved an overall accuracy of 96.7% on the test set.

Lu et al. [24] explored the use of CNN for the classification of rice diseases. They used 500 images captured in an experimental rice field to build a dataset used for training and validation purposes. AlexNet was the CNN architecture used to build a rice diseases classifier. Authors have compared the deep CNN with traditional machine learning algorithms. The overall accuracy of the deep model was 95.48% under 10-fold cross-validation. On the other side, the results of shallow models are: SVM achieved an accuracy of 91%, standard back propagation achieved 92% and Particle Swarm Optimization (PSO) achieved 88%.

Table 6.1 summarises the results of works that use DL models for plant diseases classification. We observe from Table 6.1 that most of the studies were conducted over the two last years. Also, the most used CNN architectures in these works are AlexNet, GoogleNet and similar architectures like CaffeNet.

Most of the described studies in this section focused on improving the accuracy of diseases classification without treating the interpretability challenge, by extracting insights from the classifier. There are several attempts in this direction based on visualisation methods to address this challenge. Despite these efforts, understanding and interpretation of results in the DL models is still immature and requires more attention. For example, visualisation of CNN filters as small images is used by [18, 34]. This visualisation method is applied to the first layers that interact directly with the input images. This represents a limiting factor because near-to-input layers extract only low-level feature like edges in different directions. Understanding of plant diseases still requires more abstract features like complex shapes and textures.

Visualisation of feature maps is another technique used in the classification of plant diseases [24, 25, 34]. These methods convert the internal activations to images in order to visualise the features that are activated in response to lesions and symptoms of diseases. Nevertheless, it is still difficult to localise precisely the specialised activations of these symptoms among a large number of feature maps and nodes.

The occlusion method, investigated in [5], is another method that tries to localise disease symptoms in tomato leaves. It analyses the CNN behaviour as a black box without taking into account the architecture and the internal details of CNN, such as feature maps and filters visualisations. The basic idea of this method is to apply occlusions to some parts of the images and then observe the CNN output sensitivity regarding to these occlusions. The advantage of this method is its ability to determine which image parts are important from the CNN viewpoint. However, this method is sensitive to hyperparameters like the shape, the size and the displacement stride of occlusion regions. Furthermore, a large number of occluded images are used as input of CNN which makes it computationally expensive and time-consuming.

6.3 Comparison Between Deep and Shallow Approaches

DL represents a new promising trend in the classification of plant diseases. Recently, DL algorithms have achieved the state-of-the-art in many domains, particularly in computer vision, by giving spectacular results compared to classic machine learning algorithms. For instance, the top 5 classification error achieved by the deep network called AlexNet in ImageNet dataset is 15.3% whereas the classic machine learning algorithms have a top 5 error of 26.2%. Likewise, in plant diseases classification DL outperformed shallow classifiers results and recently became a hot topic [5, 25].

These DL algorithms are different from classic machine learning algorithms in the following points:

Data Consumption: The supervised training of DL classifiers requires a large number of labelled examples, for this reason, data availability in the last decade has contributed to DL success [6]. DL classifiers require a huge training set because these classifiers contain a large number of parameters to tune. This constraint of labelled data represents a limiting factor when the labelling is expensive. For example, the biological labelled examples are expensive and difficult to collect in most cases [3]. Plant diseases classification is an example of a biological field where data collection and labelling is very expensive. In this context, each image must be labelled by an agriculture expert who should have an academic knowledge supported by practical experience in identification of disease symptoms. Also, taking a large number of images containing many diseases represents a tedious task and many years are required to cover all the existing diseases. Various approaches have been proposed to handle this data avidity. Pre-training a deep architecture with a big labelled dataset like ImageNet and fine-tuning this pre-trained architecture using a small dataset is used in many studies [25]. Moreover, data augmentation can help in increasing the number of labelled examples and variations in the training set [34]. Despite that, the adaptation of DL to domains where labelling is expensive represents an active research area and requires more effort [19, 28, 31, 38].

Dedicated Hardware: The training phase of DL classifiers requires dedicated hardware like the Graphics Processing Units (GPUs) to reduce execution time [21]. These GPUs represent an essential component in DL approaches and training without GPUs leads to many further days of training. However, the testing phase of a DL classifier does not require any dedicated hardware and can be executed on small devices like mobiles or embedded systems. In plant diseases classification, a DL classifier is trained in dedicated hardware and once the training is finished, this classifier is deployed to users on mobiles to detect diseases directly in the fields.

Feature Extraction: Machine Learning algorithms contain a feature engineering phase. In this phase, experts propose the hand-crafted features to facilitate learning from examples. This phase is very important and affects the overall performance of the learning system. Unfortunately, feature engineering is a manual component in the machine learning pipeline and it is time-consuming [22]. On the other hand, in a DL pipeline, feature extraction is embedded in the learning algorithm where features are extracted in a fully automated way and without any intervention by a human expert. CNN represents a good example of automatic feature extraction in computer vision. Filters in traditional machine learning are proposed by experts of vision, where CNN filters are learned in training using a backpropagation algorithm [22]. Recently, DL features achieved better results than hand-crafted features [5, 21, 27]. Despite this superiority of DL features, they still suffer from the difficulty of interpretation. Many attempts have been made to understand the role of these features using visualisation methods. Despite these efforts, more studies are required to demystify DL features as understanding of these features is still immature [32, 39].

6.4 Deep Learning System for Plant Diseases Classification

An overview of the DL system is illustrated in Fig. 6.1. This system contains three phases presented in this section.

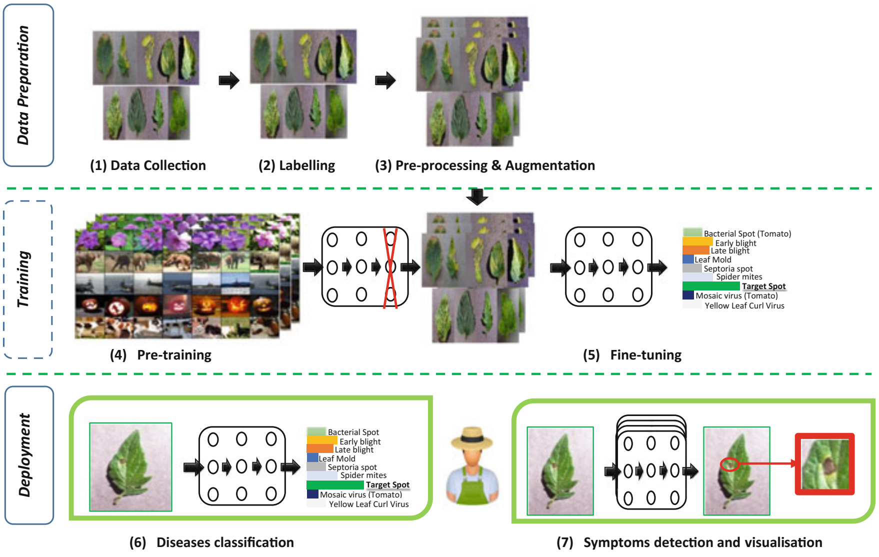


Fig. 6.1 Overview of deep learning system for plant diseases classification

6.4.1 Data Preparation

Every machine learning system starts with a data preparation phase. This data preparation phase contains the following stages:

- Data collection: is an important stage for developing any data-driven application. Particularly, deep models require a large dataset to avoid overfitting which presents a major challenge. Up to now, only a few datasets are publicly available for diseased plants. Most of the works in this area are conducted on the PlantVillage public dataset [15] or private datasets [9, 34].
- Labelling: the labelling process consists of annotating the collected images by a human expert. This expert labels images according to two possible strategies:
  - Weak labelling: where the agriculture expert identifies only the disease in each plant without any additional information about this disease.
  - Strong labelling: where the agriculture expert determines, in addition to the disease, the infected regions on the plant. This labelling strategy is expensive and time-consuming because it requires the patience of the expert where he uses the annotation software. For this reason, most of the available dataset are weakly labelled.
- Data augmentation and pre-processing: deep models like CNN are very greedy in their use of labelled data as discussed in Sect. 6.3. Unfortunately, data collection and labelling are very tedious and expensive tasks. To address this problem, data

augmentation techniques are used commonly by DL researchers. Augmentation techniques aim to increase the size of the dataset and include more variations. These techniques consist of geometrical transformations (resizing, crop, rotation, horizontal flipping) and intensity transformations (contrast and brightness enhancement, colour, noise). Moreover, image pre-processing is used to normalise the images of the dataset. The most used techniques in the DL context are image resizing and mean subtraction. The resizing is used to convert input images to the size of the network input layer. However, mean subtraction is used to centre the data which accelerate the optimisation using a gradient descent algorithm.

6.4.2 Training

After the process of data preparation, deep models and particularly CNN models are trained using backpropagation algorithm. This algorithm aims to minimize a cost function that measures the total error of the model on the training set. To reduce this error, the gradient of this cost function is calculated with respect to all weights. The gradient descent algorithm is then used to find the optimum of the cost function. For more technical details about backpropagation and gradient descent algorithms, the reader is referred to [10]. As illustrated in Fig. 6.1, the training phase contains two stages:

- Pre-training: consists of training a deep CNN on a large dataset like ImageNet first, before the training on our dataset. This pre-training is carried out to prepare the CNN by the transfer learning from a big dataset to plant diseases classification. This stage is used to deal with the lack of labelled data in plant diseases classification.
- Fine-tuning: in this stage, the last layer (output layer) of the original pre-trained network is replaced with a new layer compatible with the number of classes in our dataset. The obtained network is then retrained using the backpropagation algorithm to fit our data. This method improves the results of our model because the weights have already been trained on a bigger dataset. This fine-tuning is a transfer learning method that allows the plant diseases task to take advantage of models trained on another computer vision task where a large number of labelled images is available.

6.4.3 Deployment

The trained models can be deployed to users machines (computers, mobiles...etc.) and can be used in two modes:

- Diseases classification: a captured image is used as an input of the model, then, the output of the network determines which diseases are present in the plant.

- Symptoms detection and visualisation: the user can visualise regions that characterise the identified disease. The visualisation methods used for symptoms are very useful for inexperienced users by giving them more information about the alteration to the plant made by the disease.

6.5 Evaluation of State-of-the-Art CNN Architectures for Plant Diseases Classification Task

In this section, the state-of-the-art architectures of CNN are used for plant diseases classification based on the images of the leaves. The entire procedure is divided into several steps: preparing the dataset, training and evaluating the trained models and analysis of the results discussed in detail.

6.5.1 Dataset Structure

In order to compare our results with the existing works, the evaluation process will be conducted using the PlantVillage dataset. This dataset includes 54323 images of 14 crop species with 38 classes of diseases or healthy plants, as shown in Table 6.2.

All used images in the experimental tests are randomly cropped to be $224 * 224$ or $299 * 299$[1] according to the network input size. Only colour images are used in the training stage due to the conclusion of [25] where the results show that colour images give a better accuracy than grey scale images. Moreover, a background class containing 715 images is added in order to train the classifier to distinguish between plants leaves and the surrounding environment [34]. This class is formed using colour images from Stanford public dataset of background images [11]. Finally, the size of the final dataset after adding the background class becomes 55038 divided into 39 classes.

6.5.2 Training Deep Networks for Plant Diseases Classification

In this experiment, six state-of-the-art architectures (AlexNet [21], DenseNet-169 [14], Inception v3 [37], ResNet-34 [13], SqueezeNet-1.1 [16] and VGG13 [33]) are trained on the dataset described in the previous section. To train and evaluate the performance of these state-of-the-art CNN, we use a Python deep learning framework

[1] Images are randomly cropped to be $299 * 299$ for Inception v3 architecture and $224 * 224$ for (AlexNet, DenseNet-169, ResNet-34, SqueezeNet-1.1 and VGG13).

Table 6.2 PlantVillage dataset details

	Name	Images no
1.	Apple Scab, Venturia inaequalis	630
2.	Apple Black Rot, Botryosphaeria obtusa	621
3.	Apple Cedar Rust, Gymnosporangium juniperi-virginianae	275
4.	Apple healthy	1645
5.	Blueberry healthy	1502
6.	Cherry healthy	854
7.	Cherry Powdery Mildew, Podoshaera clandestine	1052
8.	Corn Grey Leaf Spot, Cercospora zeae-maydis	513
9.	Corn Common Rust, Puccinia sorghi	1192
10.	Corn healthy	1162
11.	Corn Northern Leaf Blight, Exserohilum turcicum	985
12.	Grape Black Rot, Guignardia bidwellii	1180
13.	Grape Black Measles (Esca), Phaeomoniella aleophilum, Phaeomoniella chlamydospora	1383
14.	Grape Healthy	423
15.	Grape Leaf Blight, Pseudocercospora vitis	1076
16.	Orange Huanglongbing (Citrus Greening), Candidatus Liberibacter spp.	5507
17.	Peach Bacterial Spot, Xanthomonas campestris	2297
18.	Peach healthy	360
19.	Bell Pepper Bacterial Spot, Xanthomonas campestris	997
20.	Bell Pepper healthy	1478
21.	Potato Early Blight, Alternaria solani	1000
22.	Potato healthy	152
23.	Potato Late Blight, Phytophthora infestans	1000
24.	Raspberry healthy	371
25.	Soybean healthy	5090
26.	Squash Powdery Mildew, Erysiphe cichoracearum	1835
27.	Strawberry Healthy	456
28.	Strawberry Leaf Scorch, Diplocarpon earlianum	1109
29.	Tomato Bacterial Spot, Xanthomonas campestris pv. vesicatoria	2127
30.	Tomato Early Blight, Alternaria solani	1000
31.	Tomato Late Blight, Phytophthora infestans	1591
32.	Tomato Leaf Mould, Passalora fulva	1909
33.	Tomato Septoria Leaf Spot, Septoria lycopersici	952
34.	Tomato Two Spotted Spider Mite, Tetranychus urticae	1771
35.	Tomato Target Spot, Corynespora cassiicola	1676
36.	Tomato Mosaic Virus	1404
37.	Tomato Yellow Leaf Curl Virus	373
38.	Tomato healthy	5375

Table 6.3 Machine characteristics

No	Hardware and software	Characteristics
1.	Memory	16 Gb
2.	Processor (CPU)	Intel Core i7-4790 CPU @ 3.6 GHz x8
3.	Graphics (GPU)	GeForce GTX TITAN X 12 Gb
4.	Operating system	Linux Ubuntu 16.04 64 bits

called pyTorch with a GPU acceleration option[2]. Our pytorch implementation is available at https://github.com/MarkoArsenovic/DeepLearning_PlantDiseases.

These six CNN architectures are trained for the plant diseases classification task using three different strategies. Two of these strategies are based on the transfer learning from pre-trained networks. The first transfer learning approach, called shallow strategy, consists of fine-tuning only the fully connected layers, while the rest of the network is used as a feature extractor. On the other hand, the second transfer learning strategy, called deep strategy, fine-tunes all network layers and starts backpropagation optimisation from the pre-trained network. Using these two approaches, the CNN classifier tries to learn more specific features for plant diseases classification starting from pre-trained networks. Finally, the third strategy consists of training the CNN from scratch starting from a random configuration of weights.

All these 18 training configurations (6 CNN architectures × 3 strategies) use the same hyperparameters values (momentum 0.9, weight decay 0.0005, learning rate 0.001, batch sizes 20). The dataset is divided into 80% for training and 20% for evaluation. All experiments are performed on a powerful machine, having the specifications that are summarized in in Table 6.3.

6.5.3 *Model Evaluation Results and Discussion*

The accuracy and training time for all the six CNN architectures using the different training strategies for plant diseases classification task is displayed in Table 6.4.

The obtained results can be compared to [25], where they used the same dataset, except for the background class added in our experiments. All other works in plant diseases classification, described in related works, used only part of the PlantVillage dataset specific to particular plant species [5] or their privately collected datasets [8, 9, 18, 24, 27, 34].

In [25], authors used only two architectures AlexNet [21] and GoogLeNet (Inception v1) [36] and based on two learning strategies: training from scratch and transfer learning. In this study, the accuracy results of the training from scratch, using 80–20% train-test distribution, are (AlexNet, 97.82%) and (GoogleNet, 98.36%), while

[2] https://github.com/pytorch/pytorch.

Table 6.4 Experiment results

Model	Training type	Training time [h]	Accuracy
AlexNet	Shallow	0.87	0.9415
AlexNet	From scratch	1.05	0.9578
AlexNet	Deep	1.05	0.9924
DenseNet169	Shallow	1.57	0.9653
DenseNet169	From scratch	3.16	0.9886
DenseNet169	Deep	3.16	0.9972
Inception_v3	Shallow	3.63	0.9153
Inception_v3	From scratch	5.91	0.9743
Inception_v3	Deep	5.64	**0.9976**
ResNet34	Shallow	1.13	0.9475
ResNet34	From scratch	1.88	0.9848
ResNet34	Deep	1.88	0.9967
Squeezenet1_1	Shallow	**0.85**	0.9626
Squeezenet1_1	From scratch	1.05	0.9249
Squeezenet1_1	Deep	2.1	0.992
VGG13	Shallow	1.49	0.9223
VGG13	From scratch	3.55	0.9795
VGG13	Deep	3.55	0.9949

the accuracy results of training using transfer learning are (AlexNet, 99.24%) and (GoogleNet, 99.34%).

From Table 6.4, Inception v3 network (modification of GoogLeNet by introducing batch normalisation [37]) gives the best accuracy for deep training strategy, with 99.76% outperforming the results shown on [25].

From the results of Table 6.4, we observe that the most successful learning strategy in the classification of plant diseases for all CNN architectures is the deep transfer learning. Also, we can observe that DenseNet169 has a comparable accuracy to Inception-V3 with 99.72% but with less training time, followed by ResNet34 that has an accuracy of 99.67% with even less training time. Furthermore, VGG13, using deep transfer learning strategy, is ranked fourth according to accuracy with 99.49%. AlexNet and SqueezeNet have a similar accuracy of 99.2% which is smaller than the results of other architectures. DenseNet169 gave the best accuracy in the other two strategies (shallow and from scratch) with 98.86 and 96.53% respectively.

Evaluation of the performance of the models by comparing training time and accuracy is also displayed in Fig. 6.2.

A large fraction of the PlantVillage dataset are images of leaves in a controlled environment and simple background. Adding images with different qualities and complex backgrounds in the training and validation dataset could improve accuracy and produce a classifier more useful for practical usage. The PlantVillage dataset

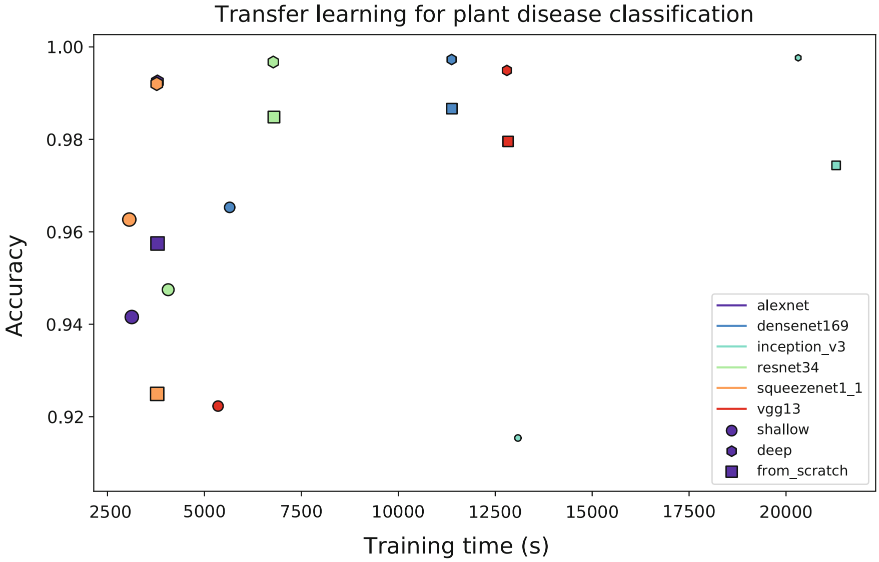


Fig. 6.2 Training time and accuracy of CNN architectures

is unbalanced, where some classes have more images than others, which could be very misleading and could lead to overfitting if not trained carefully. Augmentation techniques could help in these situations, and it is a common procedure in many classification tasks.

6.6 Deep Learning Visualisation Methods

Despite the good results of DL classifiers, they are often considered as black boxes because of their lack of interpretability. The superposition of layers and the use of nonlinear functions make the understanding of the classification difficult. Thus, DL classifiers require specialized algorithms to deal with this interpretability challenge, by extracting insights from the classifier [32, 35, 39].

These visualisation algorithms help both the designer and the user of the classifier. The classifier designer uses these algorithms to analyse the classifier behaviour to improve the performance, while the user benefits from the transparency offered by the visualisation in order to understand the classification. In many domains, classification is insufficient and requires support by an explanation of this automatic decision. Notably, plant diseases classification represents a domain where the classifiers understanding is very important. For users, a visualisation algorithm helps in disease understanding by localising the symptoms and the infected regions. Hence, biological knowledge is extracted from the classifier to help the non-expert farmers, while agriculture experts and classifier designers use visualisation to understand the classifier behaviour [5]. For example, a classifier having a good accuracy may use contextual information in the image to detect the disease and ignore symptoms. In this situation, the classifier will suffer from a generalisation problem if the subject background is different. Without visualisation algorithms, it is difficult to identify this issue based on the classifier performance. In practice, this situation of classification by background context may happen, if all training images of one disease are taken in the same background by one farmer.

In literature, there are many proposed visualisation algorithms for DL classifiers. In this chapter, we focus on algorithms proposed for CNN. This choice is motivated by the extensive use of CNN for images and particularly in plant diseases detection.

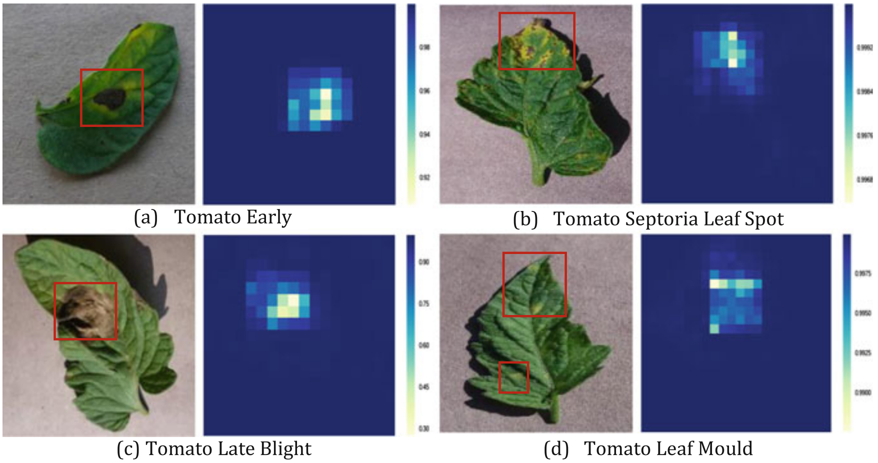
6.6.1 Visualisation in Input Image

This type of methods is very important from the practical viewpoint because it projects the features used by network back to the input image. Therefore, the image can be examined to understand how the classifier behaves [32, 35]. In plant diseases, these visualisations give valuable information about the important parts used by the network as features. If the classifier behaves correctly, these parts may represent the symptoms or the characteristics of a disease [5]. However, if the classifier uses the background or another feature unrelated to disease, then this undesirable behaviour can be detected [5].

6.6.1.1 Occlusion Experiments

Occlusion experiments aim to analyse the network sensitivity to the occlusions of image regions. Using this method, the classifier designer examines whether the network captures the characteristics of the image or not during the classification. For instance, the background of an infected leaf in a plant should not affect the diagnosis of the disease. However, the classification should be based on the symptoms of each disease in the leaf. In this specific situation, occlusion of the leaf background should not affect the classifier decision in the same way of an occlusion of a symptom region [5].

For implementing occlusion experiments, a black square is used to occlude a region in the input image. The obtained image is then classified by the network to produce an output vector. This vector is examined to understand the sensitivity of the network to this occlusion. Precisely, the node corresponding to the ground truth class of the image is checked. Naturally, if this region is important, then the value of the mentioned node decreases dramatically, in response to occlusion of this region. However, if the occluded region is not important then the node value, corresponding to ground truth class, does not fluctuate very much [39].



(a) Tomato Early

(b) Tomato Septoria Leaf Spot

(c) Tomato Late Blight

(d) Tomato Leaf Mould

Fig. 6.3 Occlusion experiments in plant diseases classification. The red squares drawn in the leaves images represent the most active occluded parts in the heat map

The square of occlusion is slided by stride over the image to produce a heat map formed using the ground truth node values. This heat map visually demonstrates the influence of each region on the classification. Hotter regions are likely to be important in classification and colder regions are likely to be less important. Similarly, users of the classifier can benefit from this method to understand the disease and its symptoms by viewing the importance of each image region [39].

Figure 6.3 shows the tendency of the network to focus on lesions caused by the diseases while ignoring the healthy part of the leaf in addition to the background. Specifically, the heat map of Fig. 6.3a indicates precisely the location of grey concentric rings in the infected leaf. The occlusion of this symptom affects the classifier decision more than the leaf regions of the other samples. This result is compatible with the experts defined symptoms for early blight [20]. Similarly, for Septoria, Late Blight and Leaf Mould, active regions in the heat map match exactly the lesions that characterise these diseases. Nevertheless, the heat map visualisation misses some infected regions in the leaf. For instance, in Fig. 6.3d some yellow regions in the bottom part of the leaf are not shown on the heat map.

Occlusion experiments suffer from some problems and have many drawbacks. To produce a heat map, a large number of occluded images are used as input to the network which makes it computationally expensive and time-consuming. Considering an image of $500 * 500$ pixels resolution and an occlusion square having the size $50 * 50$ pixels and slided by a stride of 10 pixels, the size of the produced heat map is $46 * 46$ which requires 2116 occluded images. Similarly, if the stride is only one pixel, then the number of occluded images grows exponentially to $451 * 451 = 203401$ images. In addition to the computation cost, occlusion experiments are inefficient if there are several important regions in one image. In this case, the occlusion of one region among these regions does not affect so much of the network decision. This situation is likely to occur in plant diseases classification if the symptom of a disease is dispersed overall the leaf.

6.6.1.2 Saliency Map

As mentioned above, occlusion experiments are computationally expensive and time-consuming thus, another method for estimating the importance of image regions is a necessity. Notably, a saliency map is an analytical method that allows estimation of the importance of each pixel, using only one forward and one backward pass through the network [32].

The intuition behind this method is that, if one pixel is important in respect to the node corresponding to ground truth y, then changing the values of this pixel leads to a big change in this latter node. Therefore, if the value of the gradient in this pixel is big with absolute value, then this pixel is important. Conversely, if the gradient is equal or close to zero, then the pixel is not important and its variations do not affect the output node corresponding to y. The aggregation across the channels is achieved in order to estimate the overall importance of pixels rather than the importance of each pixel channel alone.

We can consider this method as an analytical version of occlusion experiments. In occlusion experiments, the saliency map matrix is estimated numerically by modifying pixels and observing the output changes. For this reason, the calculation of a saliency map is not computationally expensive like the calculation of the heat map in occlusion experiments, since the calculation of gradient in a numerically discreet way requires the modification of each pixel or region in the image in order to approximate his gradient. However, calculating gradient analytically requires only one backward pass to calculate all the derivatives with respect to all pixels.

To the best of our knowledge, the saliency map has never been used in plant disease classification. The utility of a saliency map is comparable to occlusion experiments. It helps to identify the symptoms of diseases for users. Moreover, this method is insensitive to the dispersed important regions, because the importance of pixels is calculated analytically and is not based on occluding pixels.

To calculate the saliency map, the input image x is forwarded through the network to calculate the output of network noted $f(x)$. Then, a backward pass is used to calculate the gradient of $f(x)_y$ with respect to the input image x where y is the ground truth label corresponding to the input image x. More formally, the gradient $G(x)$, using formula (6.1), is calculated to estimate the importance of each pixel in the image x. This $G(x)$ is a tensor having the same dimension of the image x. If x has a width W, height H and three channels, then $G(x)$ is a tensor having the dimension $3 * W * H$ and indexed by three indexes: i for indexing channels and j, k for indexing pixels.

$$G(x) = \frac{df(x)_y}{dx} \tag{6.1}$$

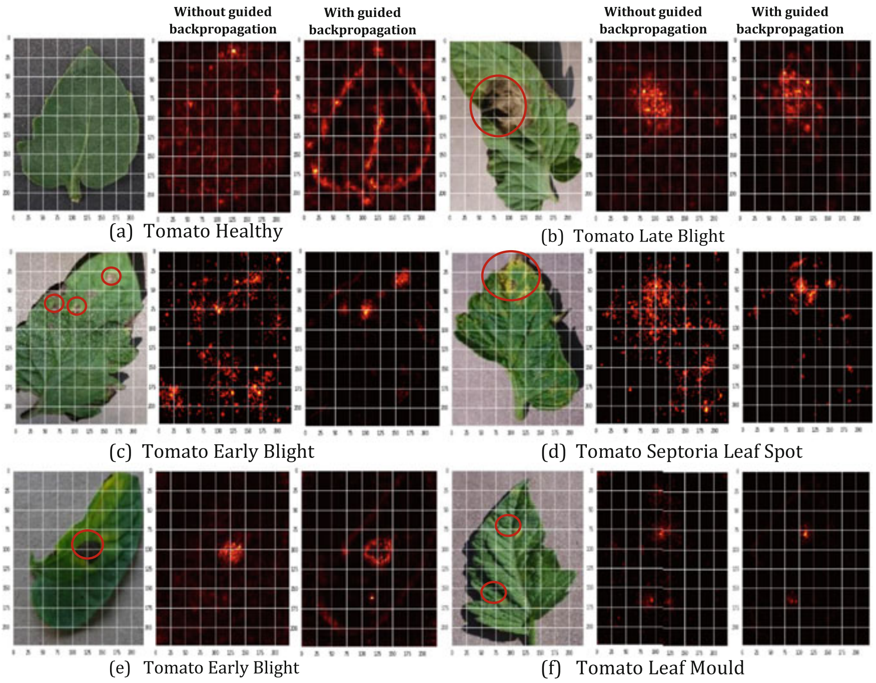

Without guided backpropagation
With guided backpropagation
Without guided backpropagation
With guided backpropagation
(a) Tomato Healthy
(b) Tomato Late Blight
(c) Tomato Early Blight
(d) Tomato Septoria Leaf Spot
(e) Tomato Early Blight
(f) Tomato Leaf Mould

Fig. 6.4 Saliency map in plant diseases classification. For image leaves in column 1, 4 there are two types of visualisations. Images in column 2, 5 represent visualisations without guided backpropagation. Images in column 3, 6 represent a visualisation using guided backpropagation

To estimate the importance of a pixel $x(i, j)$, the maximum of the absolute values across channels is calculated. Consequently, the produced matrix having the dimension $W * H$ is called saliency map SM and calculated using the following formula:

$$SM(i, j) = Max\,\{|G(0, i, j)|, |G(1, i, j)|, |G(2, i, j)|\} \tag{6.2}$$

The saliency map can localise with a good precision the infected regions in the input leaf image. Fig. 6.4b, e, f represent good examples where the visualisation of the saliency maps labels exactly the infected regions in leaves. Moreover, in Fig. 6.4f the two dispersed regions of the leaf mould disease are localised in contrast with the occlusion experiments that show only one infected region. Despite these good results, in many cases the saliency maps are not clear and suffer from noisy activations that can disturb the user. As an example, the visualisations in Fig. 6.4c, d show many activated regions in addition to the infected ones. This limitation is addressed by adding the guided backpropagation option which is described in the next section.

6.6.1.3 Saliency Map with Guided Backpropagation

The guided backpropagation method adds an additional rule during the backward pass. This rule is applied during the backpropagation through the nonlinear function called rectified linear (ReLU). In contrast with the standard backpropagation, only positive gradients are backward through ReLU [35]. This rule prevents the backward flow of negative gradients on ReLU from the higher layer in the CNN architecture [35]. This stops the gradients originated from the neurons that decrease the activation of the class node $f(x)_y$ and keeps the gradients from neurons that increase the activation of class node $f(x)_y$. Interestingly, unlike the standard backpropagation, this method produces more precise visualisations which help the user in detection of infected regions [35].

As shown in Fig. 6.4, the noisy activations on Fig. 6.4c, d are filtered and the visualisation become sharper. The three infected regions of early blight are now clear and easily distinguishable. Similarly, the main infected region of Septoria Leaf Spot is clear which gives the user a good intuition about this disease.

Furthermore, as illustrated in Fig. 6.4a, guided backpropagation produces a nice visualisation for healthy leaves. In this specific case, the network detects the contour of the leaf because no symptom is available in the leaf. This result shows the power of the network in understanding the input image by focalising only in the regions of interest and ignoring the background and non-infected regions.

6.7 Plant Diseases Classification Challenges

6.7.1 Plant Diseases Detection in Complex Images

Most studies in DL for classification of plant diseases have only focused on analysing images containing one leaf taken in a controlled environment. Although these approaches can classify a disease accurately in one image taken by a human agent, they are unable to find disease regions automatically in large fields. Obviously, a human agent is unable to monitor a large field and detects the earlier symptoms of diseases in order to take a picture. For this reason, a practical diseases detection system should automate the monitoring of fields to interact with plant diseases in due course. This automatic monitoring that leads to early detection of diseases can considerably reduce the damage on crops.

For this task, drones can fly above the field and take images for online or offline analysis. In this case, the learning system, used in data analysis, must deal with complex images containing several leaves and maybe many diseases. To achieve this, DL object detection algorithms can be used to localise and classify the affected regions in each image taken by drones. These object detection algorithms use the DL classifiers trained using simple images and adapt them to localise and classify diseases in complex images. This type of algorithm is used in literature in many contexts.

For example, in automatic driving, object detection and classification are extensively used to enhance the quality of driving. In a similar application to plant diseases classification, DL object detection algorithms are proposed in forest firefighting for detecting the fire using CNN [23]. Finally, the challenge of object detection algorithms lies in the absence of a big labelled dataset containing images taken by drones in affected crops. Efforts in this direction to collect a big dataset of labelled examples taken by drones will enhance the utility of plant diseases detection.

6.7.2 Symptoms Segmentation in Plant Diseases

DL approaches focus on diseases classification and ignores the localisation of infected regions in the leaf. This disease region identification is very important for a human expert to gain more insights into the disease and its symptoms [5]. In order to address this problem, DL segmentation algorithms can be used to divide the image into many parts in an unsupervised manner. After this segmentation, the supervised algorithm can be used to classify the disease. In this case, the user can know the diseases and the infected region at the same time. Also, more efforts in DL understanding help users to understand how classifiers classify the diseases. Visualisations algorithms should be included to enhance the user experience in understanding diseases [32, 39].

6.7.3 Labelled Images Challenge

DL algorithms require a large number of labelled images to produce a classifier not suffering from overfitting. Labelling this large number by a human expert is a real challenge and time-consuming. To address this challenge, crowdsourcing, as distributed internet framework, can produce big labelled datasets in an acceptable time. Also, crowdsourcing helps to achieve more complex labelling like the infected regions labelling [3, 30]. For example, we can divide an image using a grid of squares and ask users in crowdsourcing frameworks to select infected squares. This structured labelling can be used to evaluate automatic systems of diseases regions and symptom identification.

Crowdsourcing addresses the labelled datasets by offering an efficient tool for data collection. However, in many contexts learning algorithm must minimize the number of labelled examples and exploit unlabelled examples. Weakly and semi-supervised algorithms can be used for this objective [19, 29, 31]. This type of algorithms uses unlabelled images to improve the training of the classifier, knowing that unlabelled examples are relatively cheap and does not require human expert efforts which reduces the cost of data collection.

In addition to semi-supervised algorithms, active learning can help by choosing the examples efficiently for a labelling expert. This selective labelling improves the classifier accuracy and reduces the labelling cost at the same time [28, 38].

6.7.4 The Exploitation of Multiple Information Sources

Previous works have only focused on using images and have ignored valuable information. For example, the age of the plant is important to achieve a good disease diagnostic. Each phase in plant life has a specific set of diseases. This information can be combined with the image to avoid any incorrect decision incompatible with the age of the plant.

Climate represents an important factor for triggering plant diseases. A sudden change in temperature causes many diseases to plants. For this reason, a practical tool for protecting plants from diseases should exploit this information.

Field location is another important factor in disease detection because each region is characterised by a set of known diseases. Moreover, if one disease spreads in nearby fields, then it is likely to detect this spreading disease.

All this information can be combined with features extracted from the image to enhance the DL system performance. The fully connected layer can be combined with information from many sources and used as input of a classifier. In this case, the classifier exploits the extracted features from an image, taken directly from the field, and combines them with information that is easy to collect. Commonly, the climate state and location information can be retrieved easily from the internet on smartphones. Exploitation of this cheap information allows the detection system to adapt to the environmental factors in real time.

6.8 Conclusion

We studied in this chapter a recent trend for building a system able to detect and classify plant diseases. After analysing and comparing the previous work based on DL, we have concluded that these studies principally use two CNN architectures (AlexNet and GoogleNet). For this reason, we have evaluated the state-of-the-art CNN architectures using a public dataset of plant diseases. The results of this evaluation show clearly that we can improve the accuracy using a new CNN architecture such as inceptionV3 which achieved an accuracy of 99.76%.

In addition to this improvement in accuracy, we have investigated increasing the transparency of deep models based on visualisation techniques. In this context, the saliency map method is introduced for localising the infected regions of the plant after the identification of diseases. Despite the fact that the training images are weakly labelled, this visualisation has succeeded in the extraction of the infected regions without any expert intervention. Furthermore, this visualisation method shows a precise and sharp visualisation which helps the inexperienced users to understand the diseases.

As a limitation of this study, we can notice that visualisation method is not evaluated quantitatively using a defined measure. However, the images are assessed based on expert defined visual symptoms. This qualitative evaluation is motivated by the weak labelling of the dataset. Therefore, our future work will focus on the preparation of a strong labelled dataset, which makes it possible to measure numerically the performance of saliency maps visualisation.

References

1. Akhtar, A., Khanum, A., Khan, S.A., Shaukat, A.: Automated plant disease analysis (APDA): performance comparison of machine learning techniques. In: 2013 11th International Conference on Frontiers of Information Technology, pp. 60–65. IEEE Computer Society, Islamabad (2013)
2. Al Hiary, H., Bani Ahmad, S., Reyalat, M., Braik, M., ALRahamneh, Z.: Fast and accurate detection and classification of plant diseases. Int. J. Comput. Appl. **17**(1), 31–38 (2011)
3. Albarqouni, S., Baur, C., Achilles, F., Belagiannis, V., Demirci, S., Navab, N.: AggNet: deep learning from crowds for mitosis detection in breast cancer histology images. IEEE Trans. Med. Imaging **35**(5), 1313–1321 (2016). https://doi.org/10.1109/TMI.2016.2528120
4. Blancard, D.: 2 - Diagnosis of Parasitic and Nonparasitic Diseases. Academic Press, The Netherlands (2012)
5. Brahimi, M., Boukhalfa, K., Moussaoui, A.: Deep learning for tomato diseases: classification and symptoms visualization. Appl. Artif. Intell. **31**(4), 1–17 (2017)
6. Chen, X.W., Lin, X.: Big data deep learning: challenges and perspectives. IEEE Access **2**, 514–525 (2014). https://doi.org/10.1109/ACCESS.2014.2325029
7. Dandawate, Y., Kokare, R.: An automated approach for classification of plant diseases towards development of futuristic decision support system in Indian perspective. In: 2015 International Conference on Advances in Computing. Communications and Informatics, ICACCI 2015, pp. 794–799. IEEE, Kochi, India (2015)
8. DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E.L., Yosinski, J., Gore, M.A., Nelson, R.J., Lipson, H.: Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. Phytopathology (2017). https://doi.org/10.1094/PHYTO-11-16-0417-R
9. Fujita, E., Kawasaki, Y., Uga, H., Kagiwada, S., Iyatomi, H.: Basic investigation on a robust and practical plant diagnostic system. In: Proceedings - 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016, pp. 989–992 (2016)
10. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
11. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1–8. IEEE (2009)
12. Hanssen, I.M., Lapidot, M.: Major tomato viruses in the Mediterranean basin. Adv. Virus Res. **84**, 31–66 (2012)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR arXiv:1512.03385 (2015)
14. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks. CoRR arXiv:1608.0 (2016)
15. Hughes, D., Salathe, M.: An open access repository of images on plant health to enable the development of mobile disease diagnostics, pp. 1–13 (2015)

16. Iandola, F.N., Moskewicz, M.W., Ashraf, K., Han, S., Dally, W.J., Keutzer, K.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. CoRR arXiv:1602.07360 (2016)
17. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, MM '14, pp. 675–678. ACM, New York, NY, USA (2014). https://doi.org/10.1145/2647868.2654889
18. Kawasaki, Y., Uga, H., Kagiwada, S., Iyatomi, H.: Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. In: Advances in Visual Computing: 11th International Symposium, ISVC 2015, Las Vegas, NV, USA, 14–16 December 2015, Proceedings, Part II, pp. 638–645 (2015)
19. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. CoRR arXiv:1609.02907 (2016)
20. Koike, S.T., Gladders, P., Paulus, A.O.: Vegetable Diseases: A Color Handbook. Academic Press, San Diego (2007)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017). https://doi.org/10.1145/3065386
22. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
23. Lee, W., Kim, S., Lee, Y.T., Lee, H.W., Choi, M.: Deep neural networks for wild fire detection with unmanned aerial vehicle. In: 2017 IEEE International Conference on Consumer Electronics (ICCE), pp. 252–253 (2017)
24. Lu, Y., Yi, S., Zeng, N., Liu, Y., Zhang, Y.: Identification of rice diseases using deep convolutional neural networks. Neurocomputing **267**, 378–384 (2017)
25. Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. Front. Plant Sci. **7**(September), 1–7 (2016)
26. Mokhtar, U., El-Bendary, N., Hassenian, A.E., Emary, E., Mahmoud, M.A., Hefny, H., Tolba, M.F., Mokhtar, U., Hassenian, A.E., Emary, E., Mahmoud, M.A.: SVM-Based detection of tomato leaves diseases. In: Filev, D., Jabłkowski, J., Kacprzyk, J., Krawczak, M., Popchev, I., Rutkowski, L., Sgurev, V., Sotirova, E., Szynkarczyk, P., Zadrozny, S. (eds.) Advances in Intelligent Systems and Computing, vol. 323, pp. 641–652. Springer, Cham (2015)
27. Nachtigall, L.G., Araujo, R.M., Nachtigall, G.R.: Classification of apple tree disorders using convolutional neural networks. In: 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 472–476 (2016). https://doi.org/10.1109/ICTAI.2016.0078
28. Otálora, S., Perdomo, O., González, F., Müller, H.: Training Deep Convolutional Neural Networks with Active Learning for Exudate Classification in Eye Fundus Images, pp. 146–154. Springer International Publishing, Cham (2017)
29. Papandreou, G., Chen, L.C., Murphy, K.P., Yuille, A.L.: Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1742–1750 (2015)
30. Sharma, M., Saha, O., Sriraman, A., Hebbalaguppe, R., Vig, L., Karande, S.: Crowdsourcing for chromosome segmentation and deep classification. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 786–793 (2017). https://doi.org/10.1109/CVPRW.2017.109
31. Shinozaki, T.: Semi-supervised Learning for Convolutional Neural Networks Using Mild Supervisory Signals, pp. 381–388. Springer International Publishing, Cham (2016)
32. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. CoRR arXiv:1312.6034 (2013)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR arXiv:1409.1556 (2014)
34. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D.: Deep neural networks based recognition of plant diseases by leaf image classification. Comput. Intell. Neurosci. **2016** (2016)
35. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.A.: Striving for simplicity: the all convolutional net. CoRR arXiv:1412.6806 (2014)

36. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07–12 June, Boston, USA, pp. 1–9 (2015). https://doi.org/10.1109/CVPR.2015.7298594
37. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826. arXiv:1512.0 (2015)
38. Wang, K., Zhang, D., Li, Y., Zhang, R., Lin, L.: Cost-effective active learning for deep image classification. CoRR arXiv:1701.03551 (2017)
39. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics. LNCS, vol. 8689 (PART 1), pp. 818–833 (2014)