Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: 0005-1144 (Print) 1848-3380 (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

Automated disease classification in (Selected) agricultural crops using transfer learning

Krishnaswamy Rangarajan Aravind & Purushothaman Raja

To cite this article: Krishnaswamy Rangarajan Aravind & Purushothaman Raja (2020) Automated disease classification in (Selected) agricultural crops using transfer learning, Automatika, 61:2, 260-272, DOI: 10.1080/00051144.2020.1728911

To link to this article: https://doi.org/10.1080/00051144.2020.1728911

Published online: 24 Feb 2020.

Submit your article to this journal

Article views: 600

View related articles

View Crossmark data

Citing articles: 1 View citing articles

AUTOMATIKA
2020, VOL. 61, NO. 2, 260–272
https://doi.org/10.1080/00051144.2020.1728911

Taylor & Francis
Taylor & Francis Group

REGULAR PAPER

Automated disease classification in (Selected) agricultural crops using transfer learning

Krishnaswamy Rangarajan Aravind and Purushothaman Raja

School of Mechanical Engineering, SASTRA Deemed University, Thanjavur, India

ABSTRACT

The biotic stress of agricultural crops is a major concern across the globe. Especially, its major effects are felt in economically poor countries where advanced facilities for diagnosis of a disease is limited as well as lack of awareness among the farmers. A recent revolution in smartphone technology and deep learning techniques have created an opportunity for automated classification of disease. In this study images acquired through smartphone are transmitted to a personal computer via a wireless Local Area Network (LAN) for classification of ten different diseases using transfer learning in four major agricultural crops which are least explored. Six pre-trained Convolutional Neural Network (CNN) have been used namely AlexNet, Visual Geometry Group 16 (VGG16), VGG19, GoogLeNet, ResNet101 and DenseNet201 with its corresponding results explored. GoogLeNet resulted in the best validation accuracy of 97.3%. The misclassification was mainly due to Tobacco Mosaic Virus (TMV) and two-spotted spider mite. In test conditions, images were classified in real-time and prediction scores have been evaluated for each disease class. It depicted a reduction in accuracy in all models with VGG16 resulting in the best accuracy of 90%. Various factors contributing to the reduction in accuracy and future scope for improvement have been elucidated.

ARTICLE HISTORY
Received 1 July 2019
Accepted 3 February 2020

KEYWORDS
Crop disease diagnosis; automated classification; convolutional neural networks; transfer learning; smartphone

1. Introduction

Globally, different diseases are key factors affecting agricultural crop production [1]. In developing nations, the lack of knowledge in the identification of specific diseases is further complicating the control of its inducing factors. Abang et al. [2] survey on farmer's knowledge for identifying diseases in Cameroon brought to the limelight that only 21% and 16% of farmers were able to identify diseases and pests respectively. Another survey on farmer's knowledge of various diseases showed an inadequate awareness of the biotic stress in the crops [3]. Similar inadequacy has been reported by Islam, [4] for viral disease and its management. Therefore, appropriate training to the farmers can improve the identification and management of diseases. However, the feasibility of such a massive training suffers from the lack of significant experts and facilities. Moreover, each farmer cultivates different crops in their field which pose a significant challenge for the farmers to get trained as well as for the experts to train them for various diseases.

Recently, the abundance of cost-effective smartphones among the farmers has created an opportunity for classifying diseases using the images of the infected foliage [5]. The smartphone camera acts as a vision sensor which acquires color information in Red Green Blue (RGB) channels. This information along with the machine learning technologies learn to recognize the patterns of the disease based on the visible symptoms on the leaves. Traditionally features of the symptoms are represented using mathematical functions and evaluated in mapping for distinct patterns of different diseases [6–8]. These handcrafted features are fed to the machine learning algorithms such as Neural Networks (NN), Support Vector Machine (SVM), etc., for classifying the different diseases and best features which result in better accuracy were utilized [6,8,9–12]. Recently, due to the advancement in computing, a large number of artificial neurons are stacked in a specific architecture which forms deep neural networks and these are capable of learning the features automatically in contrary to the previous approach. These features are used for the image classification (in different domains) and this is popularly known as deep learning. One of the deep learning approaches namely Convolutional Neural Network (CNN) is widely used for image classification [13].

Different CNN based architectures namely AlexNet, VGG16, VGG19, GoogLeNet, ResNet, DenseNet, etc., have been developed and adopted for solving the problem of disease classification in various crops [5,9,14–19]. Training from scratch approach requires a

CONTACT Raja Purushothaman ✉ raja_sastra@yahoo.com School of Mechanical Engineering, SASTRA Deemed University, Thanjavur 613401, India

large dataset for training as it is a data-driven technique which poses a significant challenge to the researchers. Hence, an approach known as transfer learning [16] is widely used where the architecture is pre-trained with the ImageNet dataset and each processing element has optimized weights as a result of training. These pre-trained networks have been retrained again with standard datasets. For example, PlantVillage [20] includes images of common diseases from various crops using which many studies have been performed [5,9,14–19]. The performance of these pre-trained networks is (generally) excellent and in some cases, it resulted in an accuracy greater than 99%. In few other studies, authors have used their own developed datasets for the classification of various crop diseases [21–24]. Despite these efforts, the dataset for many crops is limited which is the critical input for training the deep learning-based algorithms.

This gap has motivated the authors to create a dataset for a few potential diseases in some vital crops (for which it is unavailable) based on the literature. Also, a rapid disease classification system with an easily available smartphone camera will be beneficial to the farmers. As an initial phase of this objective, a preliminary system consisting of a smartphone camera connected to a wireless Local Area Network (LAN) with a CPU will be used for classifying the disease with pre-trained deep learning architectures such as AlexNet, VGG16, VGG19, GoogLeNet, ResNet and DenseNet [25–29]. These architectures are briefly discussed in the following section. The observed results and analysis are presented in Section 3. The influential factors, drawbacks, possible solutions and future directions are included in Section 4.

2. Materials and methods

2.1. System overview

The system consists of a smartphone, wireless LAN and a Personal Computer (PC) (as shown in Figure 1). The smartphone used in the study was Lenovo A7700 equipped with 8 Megapixel camera which costs approximately U.S. $100. Wireless LAN is used for sending the acquired images from a smartphone to the PC.

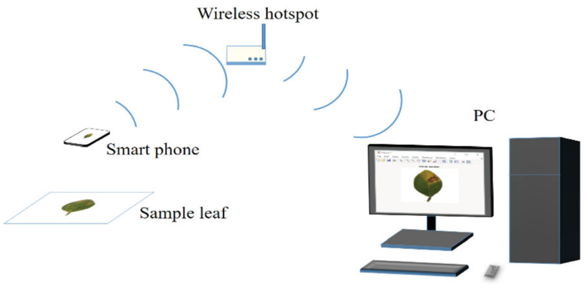


Figure 1. Overview of the system.

A mobile application (IP webcam) has been used for acquiring and transmitting the images to the PC. The computer system used for the processing of images was ACER NITRO 5 SPIN laptop with 4 GB NVIDIA 1050GTX graphics card and 8 GB of Random Access Memory (RAM). The data communication and classification utilizing deep learning was performed using MATLAB 2018a.

2.2. Dataset and deep learning models

In this study, ten different diseases (as shown in Figure 2) of four variety of crops namely eggplant (*Solanum melongena*), hyacinth beans (*Dolichos lablab*), ladies finger (*Abelmoschus esculentus L*) and lime (*Citrus aurantifolia*) have been considered for developing a disease classification system (shown in Table 1).

The dataset was created by collecting leaf samples from the field located at Tirumalaisamudram-Thanjavur district (10°43′33.8″ N, 79°00′57.5″ E) in the state of Tamil Nadu, India. The leaves were examined by the experts and categorized into the respective disease class. A smartphone was used to acquire the image of isolated leaves placed on a white sheet of paper with glass sheet to minimize the loss of visual information from the surface. The images were acquired in a room environment with sufficient natural illumination and no special lighting was used. The images were segmented manually using the software tool to minimize the effect of the background in learning process.

The number of input images in the dataset has to be increased as deep learning models require a reasonably large dataset. Data augmentation is a standard procedure where the number of images was increased artificially by applying the image transformations such as rotation, translation and changes in intensity value which prevents a common deep learning problem of overfitting [16]. A random rotation angle within the limit of ± 30° was generated and applied to each image. In addition, a random number within the range of ± 30 pixels along the horizontal and vertical direction with a random intensity value in the range of ± 30 were used to transform each images. These limits were selected based on numerous trials and the generated dataset was used for the training and validation process. Six pre-trained deep learning models namely AlexNet, VGG16, VGG19, GoogLeNet, ResNet101 and DenseNet201 have been used for training and validation of the created dataset. AlexNet, VGG16 and VGG19 is series CNN based network (as shown in Figure 3) with repetitive convolution layer followed by nonlinear activation function namely Rectified Linear Unit (ReLU). The corresponding mathematical representation of convolution operation and ReLU are given

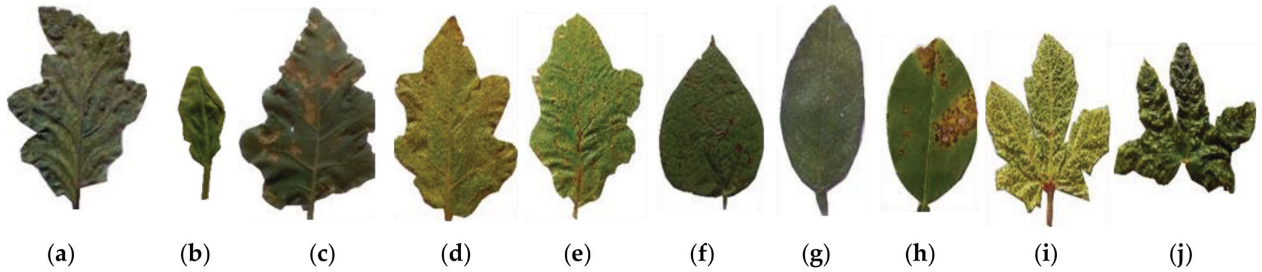


Figure 2. Disease selected for the study: (a) Tobacco Mosaic Virus TMV; (b) Little leaf disease; (c) *Epilachna* beetle; (d) Two-spotted spider mite; (e) *Cercosopora* leaf spot; (f) Brown spot; (g) Citrus hindu mite; (h) Citrus canker; (i) Yellow vein mosaic virus; (j) Leaf hopper.

Table 1. Crop disease dataset created for the study.

No	Crops	Categories	Reported damages due to the disease	No of samples	No of samples upon augmentation
1	Eggplant	(a) Tobacco Mosaic Virus(TMV)	90% yield loss [30]	96	263
		(b) Little leaf disease	40%–100% yield loss [31]	115	301
		(c) *Epilachna* beetle	80% yield loss [32]	89	310
		(d) Two-spotted spider mite	13.6%–31% yield loss [33]	71	213
		(e) *Cercospora* leaf spot	60%–80% yield loss [34]	83	231
2	Hyacinth beans	(f) Brown spot	55% yield loss [35]	82	246
3	Lime	(g) Citrus hindu mite	Reduced quality of fruit [36]	103	272
		(h) Citrus canker	Premature fruit drop & 1 % incident causes reduction in 21.3 fruits per plant [37]	108	286
4	Ladies finger	(i) Leaf hopper	50 %-63.41 % yield loss [38]	55	220
		(j) Yellow vein mosaic virus	94 % yield loss [39]	53	212
		Total		855	2554

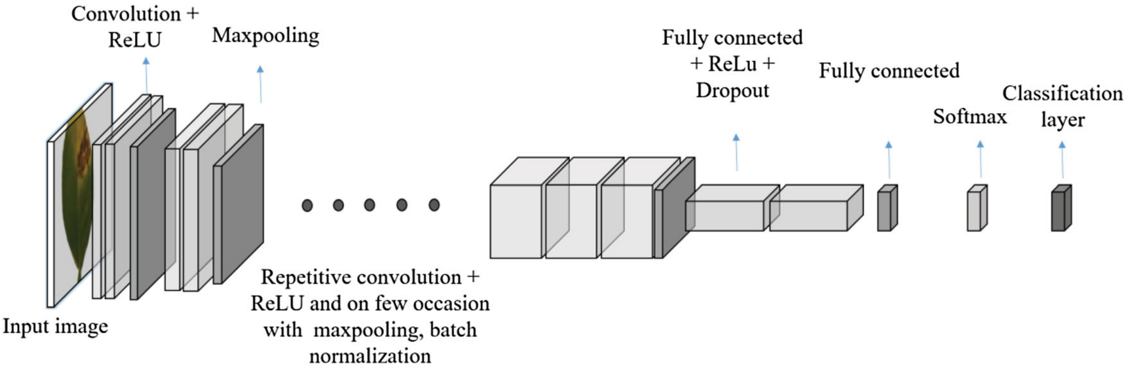

Convolution + ReLU
Maxpooling
Fully connected + ReLu + Dropout
Fully connected
Softmax
Classification layer
Input image
Repetitive convolution + ReLU and on few occasion with maxpooling, batch normalization

Figure 3. A typical architecture of AlexNet, VGG16 and VGG19.

by Equations (1) and (2) respectively [21,40].

$$X_j^l = B_j^l + \sum_{i\in M_j} K_{ij}^l * X_i^{l-1} \tag{1}$$

$$f(x) = \max(0, x) \tag{2}$$

X_j^l is the activation map from layer l as a result of convolution operation using kernel K_{ij}^l with the image or activation map of the previous layer X_i^{l-1} and bias term B_j^l. In some layers, ReLU is followed by the max-pooling (as depicted in Equation (3)) which reduces the dimension of activation map. The resulting map P_i is obtained with pooling region R_i applied on feature map (with elements α_i) from the previous layer. In case of AlexNet, addition layer known as batch normalization layer is included in few layers. The architecture finally ends with 2 fully connected layers of 4096 neurons followed by another fully connected layer with number of neurons proportional to number of classes, a softmax and a classification layer. The softmax layer (given in Equation (4)) outputs the probability y_m for the possible class with T vector of M dimension and the classification layer decides the class, based on this probability [21,40].

$$P_j = \max_{i\in R_j}\alpha_i \tag{3}$$

$$y_m = \frac{e^{T_m}}{\sum_{n=1}^{M} e^{T_n}} \tag{4}$$

GoogLeNet architecture contains inception module (as shown in Figure 4) which consists of convolution layers in parallel. It extracts features in parallel and concatenates at the end of each inception module [27]. These inception modules are arranged in a stack and

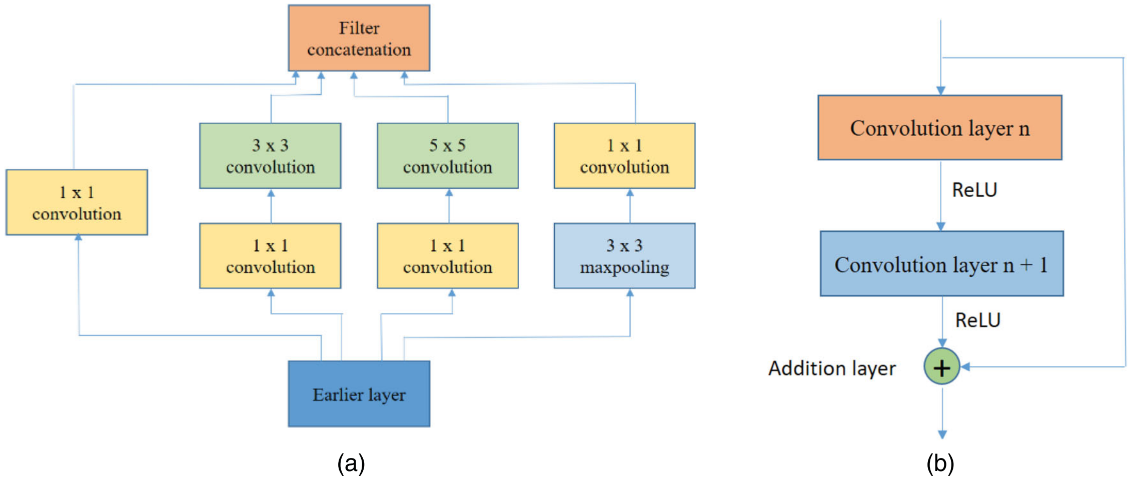


Figure 4. A model architecture of (a) Inception module used in GoogLeNet; (b) A sample residual connection.

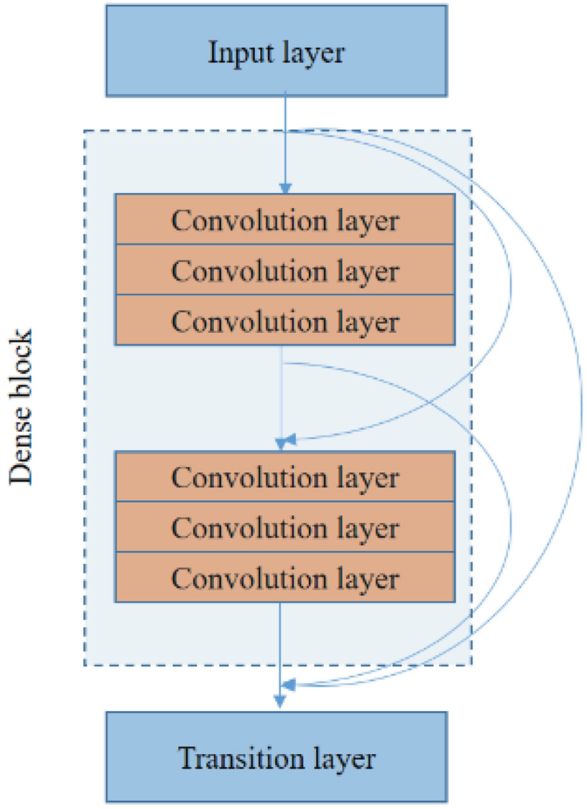


Figure 5. A model DenseNet architecture.

contain 27 layers in total. The fully connected layers which exploit the computing resource have been removed in this architecture.

As the depth of the architecture is increased by stacking more convolution layers, the accuracy and loss get saturated quickly due to vanishing gradients during backpropagation. To counteract this problem, He et al. [28] developed a technique of using a short cut or residual connections between convolution layers (as shown in Figure 4) that solves the vanishing gradient problem. There are several variants of this developed Residual Net or ResNet among which ResNet 101 (one of the deepest architectures in ResNet) were used in the study due to the limitation of the available GPU resources.

Table 2. Dataset for training, validation and testing.

Training dataset	Validation dataset	Test dataset
2044	510	50

As an alternative to the residual connection with additional layer for solving the vanishing problem, DenseNet attempts in using the concatenation layers which combine the feature maps from the previous layers instead of an additional layer (as shown in Figure 5) [29].

The architecture ensures maximum connectivity and all the layers have additional input from the previous layers. In our study, DenseNet 201 consists of 709 layers which demands comparatively higher computation resources for execution has been used.

2.3. *Methodology*

The augmented image dataset was used for training and validation (as shown in Table 2) of the six deep learning models. These trained and validated models were deployed to classify the given image (transmitted through a wireless network from a smartphone as shown in Figure 6). This sample test set was also augmented in order to increase the size and variations using which the test performance was evaluated.

During training, in order to improve the accuracy of the models, fine-tuning of the hyperparameters were performed. The following were the fine-tuned parameters for training the network: maximum epoch, 20; initial learn rate, 0.0001; L2 regularization, 0.0001 and momentum, 0.9. The following parameters were tuned for last fully connected layer: weight learning rate, 3; weight L2 factor, 1; bias learning rate, 2 and bias L2 factor, 0. Stochastic gradient descent algorithm was used for updating the weight and bias parameters. As it is a pre-trained network, the learning rate for all the layers was kept at minimum and it is required for the last fully connected layer to learn, quickly. Hence the

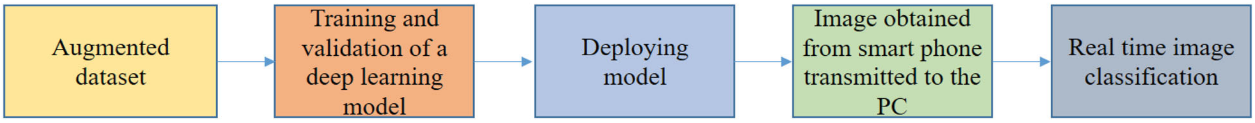
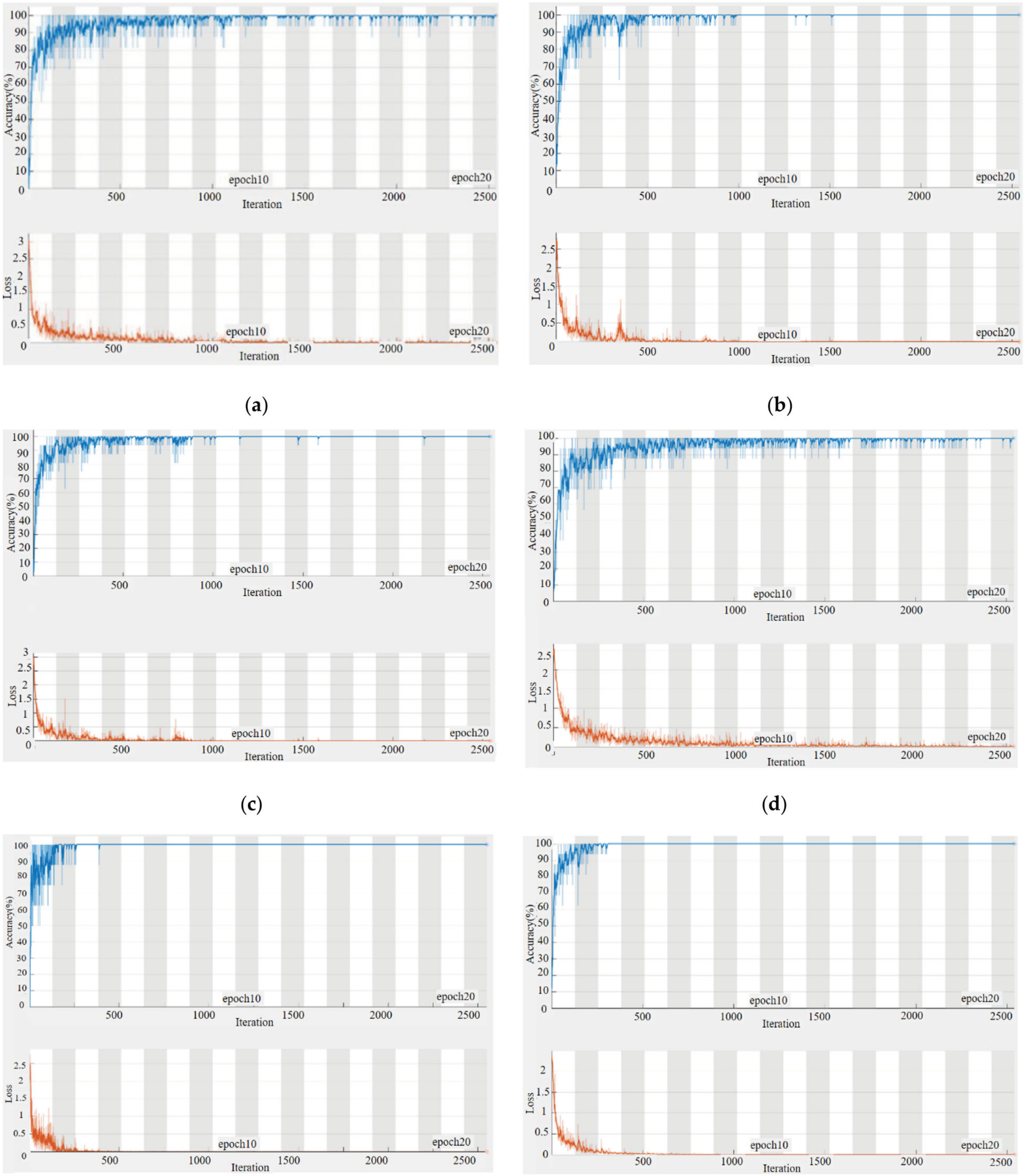

Augmented dataset
Training and validation of a deep learning model
Deploying model
Image obtained from smart phone transmitted to the PC
Real time image classification

Figure 6. Implementation.



(a)

(b)

(c)

(d)

(e)

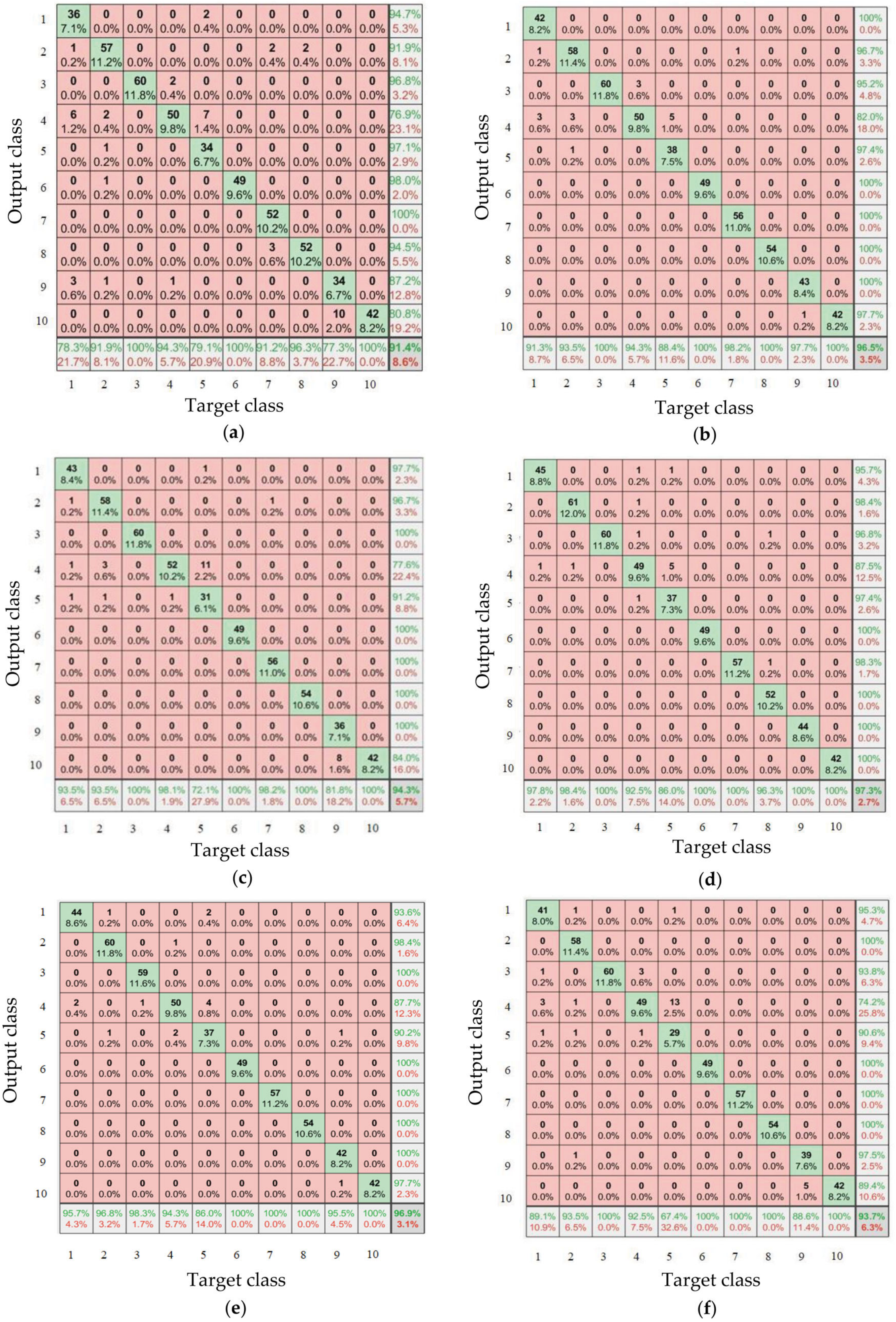
(f)

Figure 7. Training plot for (a) AlexNet; (b) VGG16;(c) VGG19; (d) GoogLeNet; (e) ResNet101; (f) DenseNet201.

learning rate for this layer is higher than the initial learning rate. The obtained results are analysed and reasons for misclassification are provided in the Results section.

3. Results

The training was carried out with 80% dataset and 20% for validation as depicted in the previous section. The



(a)

	1	2	3	4	5	6	7	8	9	10	
1	36 7.1%	0 0.0%	0 0.0%	0 0.0%	2 0.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	94.7% 5.3%
2	1 0.2%	57 11.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.4%	2 0.4%	0 0.0%	0 0.0%	91.9% 8.1%
3	0 0.0%	0 0.0%	60 11.8%	2 0.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	96.8% 3.2%
4	6 1.2%	2 0.4%	0 0.0%	50 9.8%	7 1.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	76.9% 23.1%
5	0 0.0%	1 0.2%	0 0.0%	0 0.0%	34 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.1% 2.9%
6	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	49 9.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.0% 2.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	52 10.2%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.6%	52 10.2%	0 0.0%	0 0.0%	94.5% 5.5%
9	3 0.6%	1 0.2%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	34 6.7%	0 0.0%	87.2% 12.8%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 2.0%	42 8.2%	80.8% 19.2%
	78.3% 21.7%	91.9% 8.1%	100% 0.0%	94.3% 5.7%	79.1% 20.9%	100% 0.0%	91.2% 8.8%	96.3% 3.7%	77.3% 22.7%	100% 0.0%	91.4% 8.6%

(b)

	1	2	3	4	5	6	7	8	9	10	
1	42 8.2%	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%
2	1 0.2%	58 11.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	96.7% 3.3%
3	0 0.0%	0 0.0%	60 11.8%	3 0.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	95.2% 4.8%
4	3 0.6%	3 0.6%	0 0.0%	50 9.8%	5 1.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	82.0% 18.0%
5	0 0.0%	1 0.2%	0 0.0%	0 0.0%	38 7.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.4% 2.6%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	49 9.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	56 11.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	54 10.6%	0 0.0%	0 0.0%	100% 0.0%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	43 8.4%	0 0.0%	100% 0.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	42 8.2%	97.7% 2.3%
	91.3% 8.7%	93.5% 6.5%	100% 0.0%	94.3% 5.7%	88.4% 11.6%	100% 0.0%	98.2% 1.8%	100% 0.0%	97.7% 2.3%	100% 0.0%	96.5% 3.5%

(c)

	1	2	3	4	5	6	7	8	9	10	
1	43 8.4%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.7% 2.3%
2	1 0.2%	58 11.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	96.7% 3.3%
3	0 0.0%	0 0.0%	60 11.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	1 0.2%	3 0.6%	0 0.0%	52 10.2%	11 2.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	77.6% 22.4%
5	1 0.2%	1 0.2%	0 0.0%	1 0.2%	31 6.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	91.2% 8.8%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	49 9.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	56 11.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	54 10.6%	0 0.0%	0 0.0%	100% 0.0%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	36 7.1%	0 0.0%	100% 0.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 1.6%	42 8.2%	84.0% 16.0%
	93.5% 6.5%	93.5% 6.5%	100% 0.0%	98.1% 1.9%	72.1% 27.9%	100% 0.0%	98.2% 1.8%	100% 0.0%	81.8% 18.2%	100% 0.0%	94.3% 5.7%

(d)

	1	2	3	4	5	6	7	8	9	10	
1	45 8.8%	0 0.0%	0 0.0%	1 0.2%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	95.7% 4.3%
2	0 0.0%	61 12.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.4% 1.6%
3	0 0.0%	0 0.0%	60 11.8%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	96.8% 3.2%
4	1 0.2%	1 0.2%	0 0.0%	49 9.6%	5 1.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	87.5% 12.5%
5	0 0.0%	0 0.0%	0 0.0%	1 0.2%	37 7.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.4% 2.6%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	49 9.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	57 11.2%	1 0.2%	0 0.0%	0 0.0%	98.3% 1.7%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	52 10.2%	0 0.0%	0 0.0%	100% 0.0%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	44 8.6%	0 0.0%	100% 0.0%
10	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%	42 8.2%	100% 0.0%
	97.8% 2.2%	98.4% 1.6%	100% 0.0%	92.5% 7.5%	86.0% 14.0%	100% 0.0%	100% 0.0%	96.3% 3.7%	100% 0.0%	100% 0.0%	97.3% 2.7%

(e)

	1	2	3	4	5	6	7	8	9	10	
1	44 8.6%	1 0.2%	0 0.0%	0 0.0%	2 0.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93.6% 6.4%
2	0 0.0%	60 11.8%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.4% 1.6%
3	0 0.0%	0 0.0%	59 11.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	2 0.4%	0 0.0%	1 0.2%	50 9.8%	4 0.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	87.7% 12.3%
5	0 0.0%	1 0.2%	0 0.0%	2 0.4%	37 7.3%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	90.2% 9.8%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	49 9.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	57 11.2%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	54 10.6%	0 0.0%	0 0.0%	100% 0.0%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	42 8.2%	0 0.0%	100% 0.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	42 8.2%	97.7% 2.3%
	95.7% 4.3%	96.8% 3.2%	98.3% 1.7%	94.3% 5.7%	86.0% 14.0%	100% 0.0%	100% 0.0%	100% 0.0%	95.5% 4.5%	100% 0.0%	96.9% 3.1%

(f)

	1	2	3	4	5	6	7	8	9	10	
1	41 8.0%	1 0.2%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	95.3% 4.7%
2	0 0.0%	58 11.4%	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%
3	1 0.2%	0 0.0%	60 11.8%	3 0.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93.8% 6.3%
4	3 0.6%	1 0.2%	0 0.0%	49 9.6%	13 2.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	74.2% 25.8%
5	1 0.2%	1 0.2%	0 0.0%	1 0.2%	29 5.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90.6% 9.4%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	49 9.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	57 11.2%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	54 10.6%	0 0.0%	0 0.0%	100% 0.0%
9	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	39 7.6%	0 0.0%	97.5% 2.5%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 1.0%	42 8.2%	89.4% 10.6%
	89.1% 10.9%	93.5% 6.5%	100% 0.0%	92.5% 7.5%	67.4% 32.6%	100% 0.0%	100% 0.0%	100% 0.0%	88.6% 11.4%	100% 0.0%	93.7% 6.3%

Figure 8. Confusion matrix of (a) AlexNet; (b) VGG16; (c) VGG19; (d) GoogLeNet; (e) ResNet101; (f) DenseNet201.

Note: 1 – Cercospora leaf spot, 2 – Epilachna beetle, 3 – Little leaf, 4 – TMV, 5 – Two-spotted spider mite, 6 – Brown spot, 7 – Citrus canker, 8 – Citrus hindu mite, 9 – Leaf hopper, 10 – Yellow vein mosaic disease.

learning rate remained constant for the entire training of all the models. The training plot for all the six models is shown in Figure 7. The time taken for training of the dataset using AlexNet, VGG16, VGG19, GoogLeNet, ResNet101 and DenseNet 201 were approximately 11, 108, 224, 32, 175 and 447 min respectively. As the number of convolution layers in AlexNet is lower compared to other architectures, the time taken to complete 20 epochs is the lowest of all the models used in the study. The number of layers in GoogLeNet is broader and contains many convolution layers than VGG16 but still, it takes reasonably lower time with the utilization of minimal computational resources without sacrifice in the performance. This is mainly due to the development of architecture based on Hebbian principle [27]. The convergence of training and loss graph is faster using ResNet101 and DenseNet201.

The validation accuracy using these six models i.e. AlexNet, VGG16, VGG19, GoogLeNet, ResNet101 and DenseNet 201 were 91.4%, 96.5%, 94.3%, 97.3%, 96.9% and 93.7% respectively. The GoogLeNet with inception module found to produce the best result due to parallel processing and possibly optimal depth of the architecture which improves the learning of the features from the image. The obtained result agrees with previous studies carried out using the PlantVillage dataset [5,9] while contradicting the other study [17] where VGG16 produced best results. In general, deeper architectures such as VGG16 and VGG19 produced better results at the expense of computational resources which is evident from the time taken and memory space allotted for training. In the case of DenseNet201 and ResNet101 the accuracy did not improve significantly with deeper layers but the computing resources were used efficiently compared to the earlier models. In addition, although the time taken for training is more, test time performance is in milliseconds compared to the performance of other shallower traditional learning algorithms.

When the validation accuracy was analysed using the confusion matrix (as shown in Figure 8), it showed that the disease class which was mainly affecting the performance was two-spotted spider mite. It was mainly misclassified to TMV as the symptoms of two-spotted spider mite are not learned efficiently. Moreover, the symptoms of the above disease do not have distinct clear patterns, but an infinitesimally tiny powdery appearance due to the presence of pest on the surface of leaves. In severe cases, discoloration begins to appear which may be the possible cause to misclassify it to TMV.

This was also evident from the estimated property values namely False Negative Rate (FNR), False Positive Rate (FPR), True Positive Rate (TPR) and True Negative Rate (TNR) based on the confusion matrix as shown in Table 3.

The FPR of the class TMV is higher with all the algorithms which shows that other classes have been misclassified to TMV. The other classes affecting the accuracy were *Cercospora* leaf spot and *Epilachna* beetle which were misclassified to TMV. The above misclassification is substantiated by the TPR value of *Cercospora* leaf spot and *Epilachna* beetle with most of the models. The attributed reason may be due to the presence of discoloration and pest on the leaves in few images of *Epilachna* beetle. Few of the disease class namely brown spot, citrus canker and citrus hindu mite have highest TPR values (as shown in Table 3) for VGG16, VGG19, ResNet101 and DenseNet201 which depicts a higher prediction capability. In the case of AlexNet, the other class which majorly affected its accuracy was leaf

Table 3. Confusion matrix properties.

Class	FNR	FPR	TPR	TNR
AlexNet				
1	0.021	0.052	0.947	0.978
2	0.018	0.028	0.971	0.981
3	0	0.032	0.967	1
4	0.006	0.23	0.769	0.993
5	0.011	0.08	0.919	0.988
6	0	0.020	0.980	1
7	0.01	0	1	0.989
8	0.004	0.054	0.945	0.995
9	0.021	0.128	0.871	0.978
10	0	0.192	0.807	1
VGG16				
1	0.008	0	1	0.9915
2	0.01	0.025	0.974	0.9894
3	0	0.047	0.952	1
4	0.006	0.18	0.819	0.993
5	0.008	0.033	0.966	0.991
6	0	0	1	1
7	0.002	0	1	0.997
8	0	0	1	1
9	0.002	0	1	0.997
10	0	0.023	0.976	1
VGG19				
1	0.006	0.022	0.977	0.993
2	0.025	0.088	0.911	0.974
3	0	0	1	1
4	0.002	0.223	0.776	0.997
5	0.008	0.033	0.966	0.991
6	0	0	1	1
7	0.002	0	1	0.997
8	0	0	1	1
9	0.016	0	1	0.983
10	0	0.16	0.84	1
GoogLeNet				
1	0.002	0.042	0.957	0.997
2	0.012	0.026	0.973	0.987
3	0	0.032	0.967	1
4	0.008	0.125	0.875	0.991
5	0.002	0.016	0.983	0.997
6	0	0	1	1
7	0	0.017	0.982	1
8	0.0043	0	1	0.995
9	0	0	1	1
10	0	0	1	1
ResNet101				
1	0.004	0.063	0.936	0.995
2	0.012	0.097	0.902	0.987
3	0.002	0	1	0.997
4	0.006	0.122	0.877	0.993
5	0.004	0.016	0.983	0.995
6	0	0	1	1
7	0	0	1	1
8	0	0	1	1
9	0.004	0	1	0.995
10	0	0.023	0.976	1
DenseNet201				
1	0.010	0.046	0.953	0.989
2	0.029	0.093	0.906	0.970
3	0	0.062	0.937	1
4	0.009	0.257	0.742	0.991
5	0.008	0	1	0.991
6	0	0	1	1
7	0	0	1	1
8	0	0	1	1
9	0.010	0.025	0.975	0.989
10	0	0.106	0.893	1

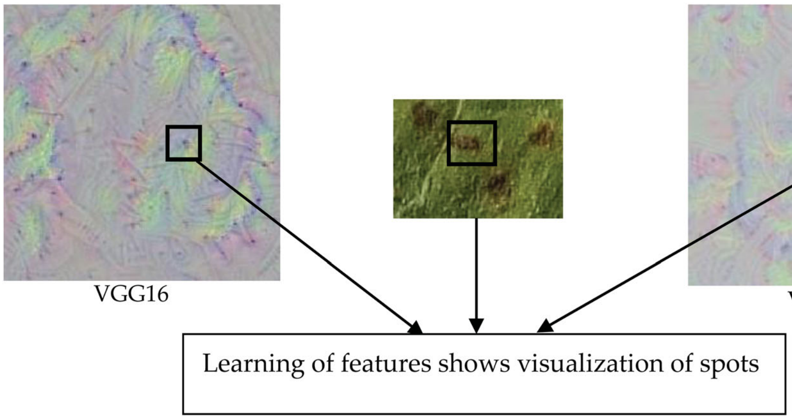


Figure 9. Comparison of sample generated image with VGG16 and VGG19 using deep dream image of *Cercospora* leaf spot.

hopper (TPR – 0.871) that was misclassified to yellow vein mosaic due to similarity in symptom color. Similar results were depicted in the case of VGG19 and DenseNet201. From the confusion matrix, it is also evident that the misclassification for disease of one species to other species was very low. It signifies the ability of architectures to learn other features such as different leaf shapes, veins, etc.

3.1. *Visualization of features*

In a further sophisticated way, the learning of features by the trained model can be visualized and analysed qualitatively using deep dream image function (originally developed by Alexnder Mordvintsevl) available in the MATLAB software tool. Few studies have used visualization for understanding the learning ability of different architecture and convolution layers [41,42]. Due to the limitation of the software tool used in this study, the function is applicable only for series networks (AlexNet, VGG16, VGG19) and it cannot be utilized for GoogLeNet as it is a Directed Acyclic Graph (DAG). The visualization gives an intuition on the learning ability of the network.

The function is applied to the last fully connected layer of AlexNet, VGG16, VGG19 as abstract features are combined by using the features from different layers. The above function generates an image which enhances the activation of the neurons in the network layers. The images were generated using the random image with pixels from a normal distribution. When the features were visualized with the AlexNet, it did not closely match the real symptoms. But, in the case of VGG16 and VGG19, features from symptoms of almost all diseases were able to be related to the visualization map. A sample visualization of *Cercospora* leaf spot is shown in Figure 9, where random spots are present in the generated image.

The poorly learned classes were not able to show features matching those disease class (e.g. two-spotted spider mite) and were generating images which may be mostly based on the weights from the pre-trained objects. This is in agreement with the analysis of the confusion matrix where two-spotted spider mite was more often misclassified to other classes.

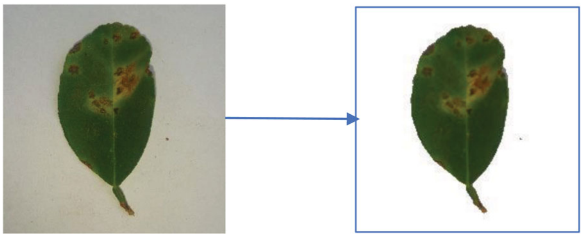


Figure 10. Sample output result obtained using the pre-processed image of class citrus canker.

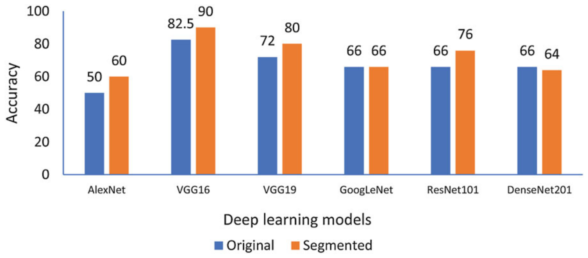


Figure 11. Test accuracy using different models.

3.2. *Testing*

Finally, testing of the sample images obtained from the smartphone camera transmitted through the wireless LAN was performed. The images were read from the specific virtual COM port of the PC with the MATLAB tool. The testing was carried out in two ways, one with pre-processing and other without pre-processing. In pre-processing a binary mask was constructed using thresholding operation and an input test image was segmented with a background similar to the training and validation images. In the other case, the input image was directly classified with the trained model. The sample output obtained as a result of using pre-processed images for classification is shown in Figure 10.

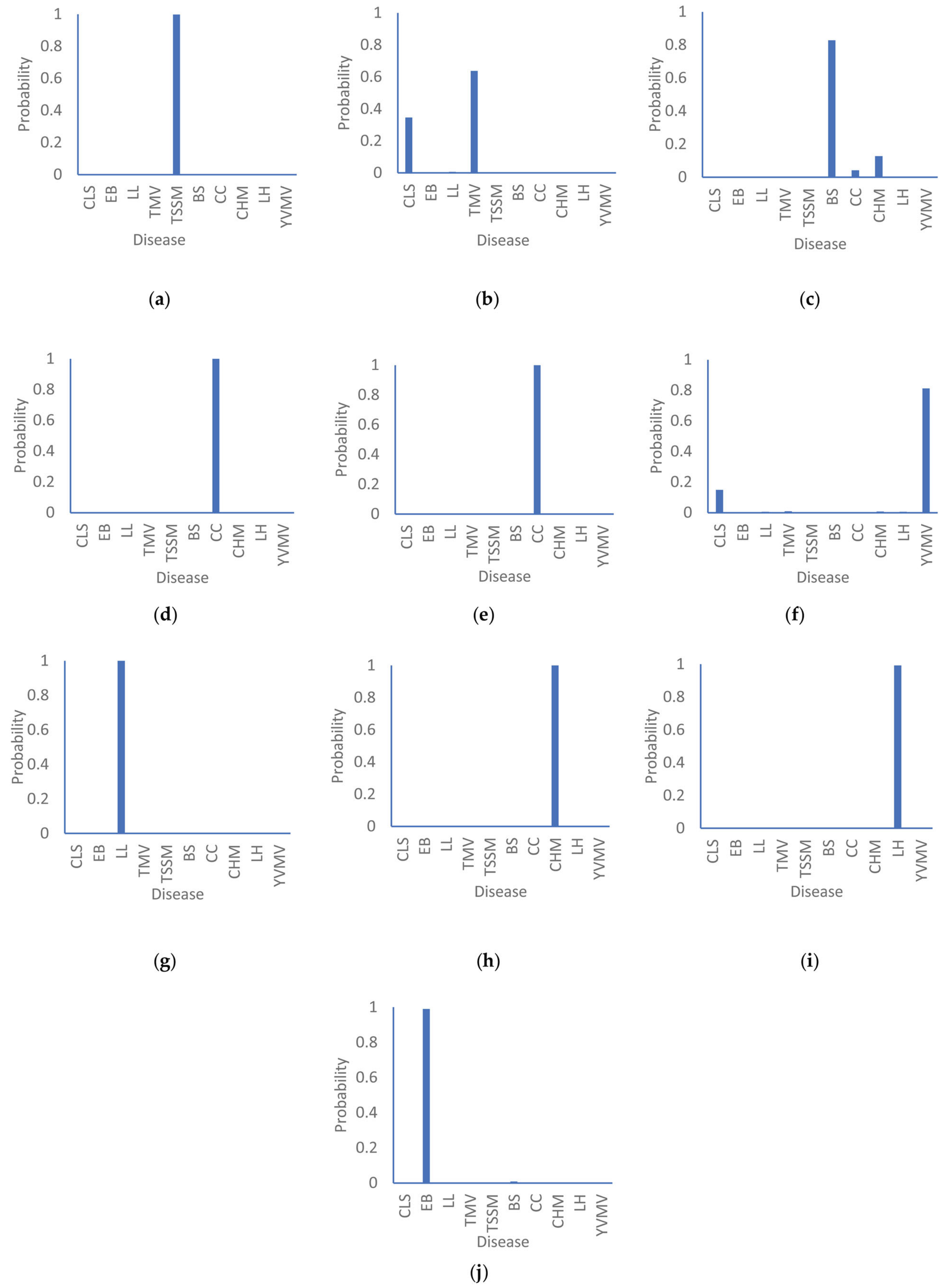


Figure 12. Prediction score using VGG16 of (a) TSSM; (b) TMV; (c) BS; (d) CC; (e) CLS; (f) LL; (g) CHM; (h) EB; (i) LH; (j) YVMV.

Note: BS – Brown spot, CC – Citrus canker, CHM – Citrus hindu mite, CLS – Cercospora leaf spot, EB – Epilachna beetle, LH – Leaf hopper, LL – Little leaf, TMV – Tobacco Mosaic Virus, TSSM – Two-spotted spider mite, YVMV – Yellow vein mosaic virus.

The prediction time of sample images for AlexNet, VGG16, VGG19, GoogLeNet, ResNet101 and DenseNet were 0.20, 0.39, 0.35, 0.27, 0.37 and 0.67 s respectively. In the test case, there was a significant difference

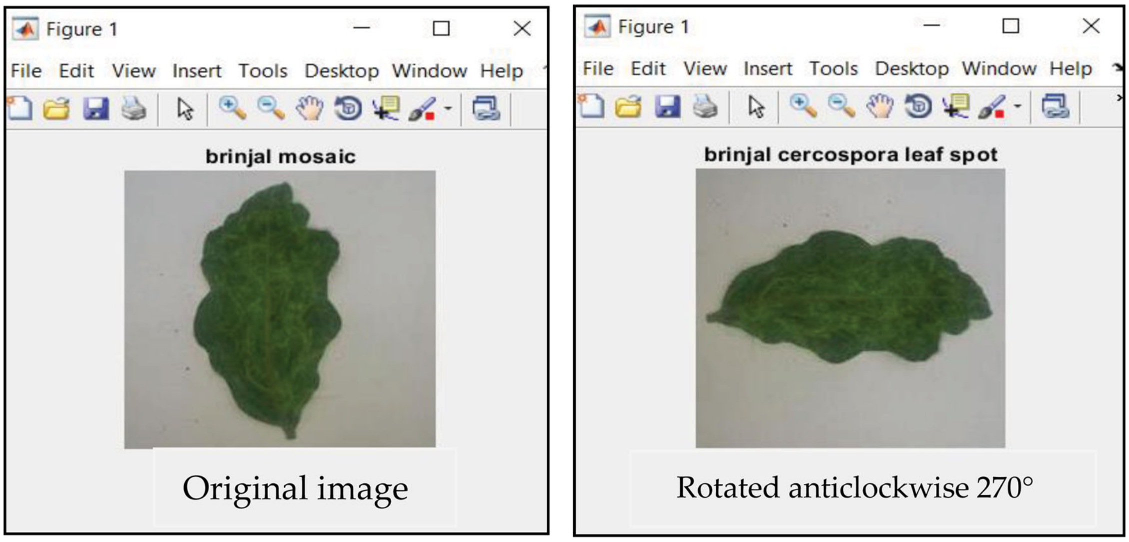


Figure 1
File Edit View Insert Tools Desktop Window Help
brinjal mosaic
Original image
Figure 1
File Edit View Insert Tools Desktop Window Help
brinjal cercospora leaf spot
Rotated anticlockwise 270°

Figure 13. Sample misclassification of TMV when the image was rotated to 270° with VGG16.

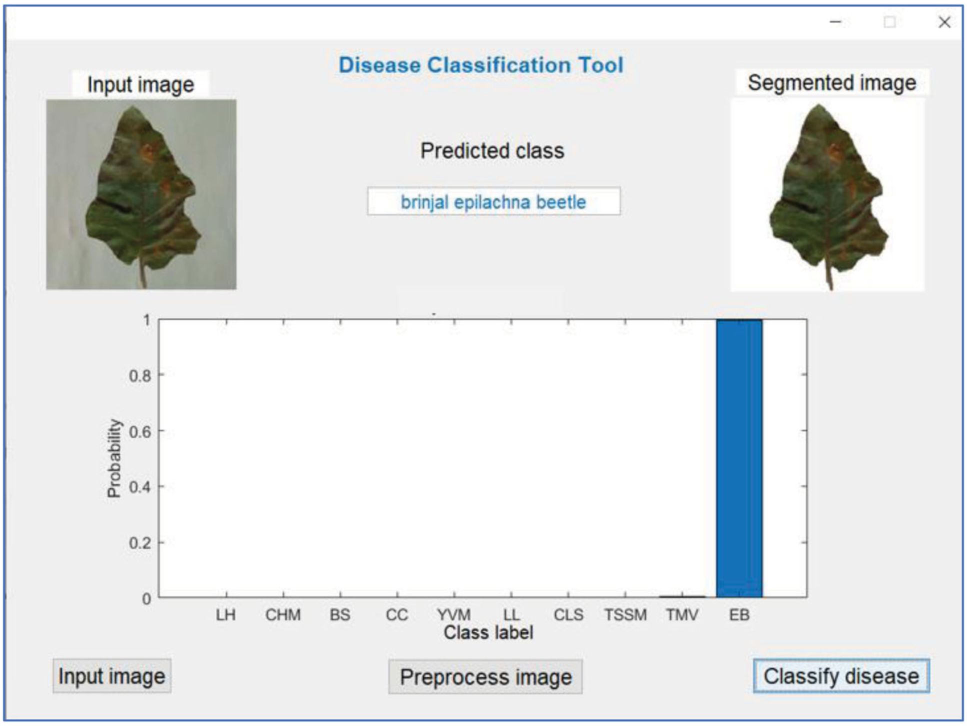


Figure 14. A Sample output result obtained using the pre-processed image of class *Epilachna* beetle.

in the classification accuracy with original and pre-processed images using different models. The applied segmentation procedure also resulted in the loss of few pixel information. As expected, the performance of AlexNet was poor with the resulting accuracy of 50% (as shown in Figure 11).

Overall, the accuracy has dropped significantly when the models were provided with the test set. It was interesting to observe that VGG16 resulted in the best accuracy (i.e. 90%) compared to other deeper architectures such as GoogLeNet (i.e. 60%), VGG19 (80%), ResNet101 (76%) and DenseNet201 (64%) using the segmented images. The class which was misclassified in almost all cases was *Cercospora* leaf spot with VGG16. It was falsely classified as citrus canker as the symptoms of this disease were similar, hence the prediction score is 100% for citrus canker as shown in Figure 12.

All the models were able to classify the six diseases with higher accuracy namely little leaf, citrus canker, citrus hindu mite, *Epilachna* beetle, yellow vein mosaic and leaf hopper. In all the models *Cercospora* leaf spot was misclassified as citrus canker as the symptoms are similar with distinct spots randomly present on the leaf surface. The prediction score for each case using the best performing model (VGG16) for the test set with pre-processed sample images for each class is shown in Figure 13.

In order to observe the effect of transformation such as rotation and translation on the test images, it was translated randomly and rotated to 90°, 180° and 270°.

There was a significant fall in the accuracy with all the models and AlexNet was poorest performer with 40% accuracy. A sample example of this above-introduced uncertainty is shown in Figure 14 where the TMV was classified as *Cercospora* leaf spot when the image was rotated to 270° with VGG16.

When an additional change in illumination was introduced, the accuracy fell further greatly as the dataset used for testing include images of untrained illumination conditions. This is in agreement with the previous studies which have shown that the deep learning architectures trained with the image acquired in controlled conditions were unable to classify the disease under different conditions effectively [5,43]. Hence, in accordance with that, the developed method is limited within the images of leaf samples acquired under constant white background with similar illumination.

4. Discussion

VGG16 alone was able to produce a classification accuracy of 90% on the test set and all the models reported a lower accuracy compared to validation. The results suggest that the trained deep learning models work well when the provided input data is similar to the training data and the accuracy drops when it is different. The effect of unseen dataset on the classification accuracy is in agreement with earlier study by Mohanty et al. [5] and Barbedo [16]. Also, the application of different augmentation techniques separately with training and test sets resulted in a drop in performance. Hence the orientation of the leaf in the input test images should be similar to the training images. The results also suggest that when the trained models were put to test without removal of the background, the accuracy fell significantly compared to segmented images in all the cases except DenseNet201. Although the deep learning model has the ability to learn features irrespective of the background, the limitation on the quantity of the dataset may be one of the factors influencing the prediction capability. Barbedo [16] however on his study showed a significant impact of the background on the classification accuracy but also suggested a solution of increasing the dataset size that may improve the performance.

In this study, *Cercospora* leaf spot in brinjal was misclassified as a citrus canker in the testing scenario. When the symptoms were analysed visually, both the disease contains a dark spot surrounded by yellow halo region. But the properties of these symptoms are different which in few cases matches each other. An earlier study shows that the similarity of the symptoms between two disease classes can lead to significant errors in classification. This problem can be partially solved by increasing the dataset size used for training according to Barbedo [16].

The studies by Ferentinos [17] and Shijie et al. [14] reported a higher accuracy of utilizing the PlantVillage dataset and VGG16. But, one of the studies by Too et al. [19], reported a lower accuracy of 76.12% using VGG16 and PlantVillage dataset which is contradicting the results of these studies. Despite the higher performance shown by VGG16, the major factor acting as a bottleneck for its implementation is the heavy exploitation of computer resources even with shallow depth compared to other deeper architectures.

Most of the studies have utlitised the PlantVillage dataset but unfortunately, it lacks several unexplored diseases and crops [5,9,14–16]. We have attempted to close this gap by the creation of a dataset for these ten diseases in four crops which is not available in the above dataset. Also, the study reveals a good accuracy, even with lower number of training images and utilization of limited computer resources for training compared to the PlantVillage dataset which has thousands of images and requires a GPU with larger memory.

Our study also has focused on application part of the system for the farmers and a Graphical User Interface (GUI) has been developed (as shown in Figure 14) using the GUI options available in MATLAB 2018a. The developed GUI is a precursor for the development of a software application that encompasses various options available to the user. When the input image option is selected, a snapshot of the image is acquired from video footage of the smartphone camera transmitted via wireless LAN and stored in the computer memory. Pre-processing is an optional step using which the image can be segmented. Finally, the classify disease option loads one of the trained models and it predicts the associated disease class. The output of the application is the predicted class with prediction score. The prediction score reveals the confidence of the network in predicting the particular target class. Although the developed system is not state of the art, it can be utilized as disease diagnostic support system in remote areas which lacks facilities and manpower for diagnosis.

In future, it is proposed to prepare a dataset of the above diseases with complex backgrounds under varying conditions. The behavior of the architectures will be analysed for developing a sustainable disease diagnostic system for the real-time classification of disease by implementing a model in smartphone.

5. Conclusions

The above study has proposed an automated disease diagnostic system with cost-effective resources such as smartphones, wireless network and a PC using intelligent deep learning-based algorithms, for assisting the farmers. The study has evaluated six pre-trained deep learning models namely AlexNet, VGG16, VGG19, GoogLeNet, ResNet101 and DenseNet201 for classification of disease, in the created dataset of ten different diseases for four different crops. The data augmentation technique has been utilized for enhancing the dataset with the introduced distortion which will improve generalization and prevents overfitting problems. When the results from the validation data were compared, GoogLeNet was found to produce the best accuracy of 97.3%. But contrastingly, in the case of test scenario, VGG16 resulted in the best accuracy. The reasons for variation in accuracy resulting from validation and test set were discussed and future direction for the above work has been presented.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Savary S, Ficke A, Aubertot JN, et al. Crop losses due to disease and their implications for global food production losses and food security. Food Sec. 2012;4(4):519–537. doi:10.1007/s12571-012-0200-5.

[2] Abang AF, Kouame CM, Abang M, et al. Assessing vegetable farmer knowledge of diseases and insect pests of vegetable and management practices under tropical conditions. Int J Veg Sci. 2014;20:240–253. doi:10.1080/19315260.2013.800625.

[3] Hasan MK, Haque ATM. Farmer's knowledge on vegetable pest and disease management. J Patuakhali Sci Tech Univ. 2014;5(1):135–143.

[4] Islam W. Management of plant virus diseases: farmer's knowledge and our suggestions. Hosts Vir. 2017;4(2): 28–33.

[5] Mohanty SP, Hughes DP, Salathe M. Using deep learning for image-based plant disease detection. Front Plant Sci. 2016;1. doi:10.3389/fpls.2016.01419.

[6] Sankaran S, Mishra A, Ehsani R, et al. A review of advanced techniques for detecting plant diseases. Comput Electron Agric. 2010;72:1–13. doi:10.1016/j.compag.2010.02.007.

[7] Oberti R, Marchi M, Tirelli P, et al. Automatic detection of powdery mildew on grapevine leaves by image analysis: optimal view-angle range to increase the sensitivity. Comput Electron Agric. 2014;104:1–8. doi:10.1016/j.compag.2014.03.001.

[8] Kamilaris A, Prenafeta-Boldu FX. Deep learning in agriculture: a survey. Comput Electron Agric. 2018;147: 70–90. doi:10.1016/j.compag.2018.02.016.

[9] Brahimi M, Boukhalfa K, Moussaoui A. Deep learning for tomato diseases: classification and symptom visualization. App Artif Intell. 2017;31:299–315.

[10] Singh V, Misra AK. Detection of plant leaf diseases using image segmentation and soft computing techniques. Inf Process Agric. 2017;4:41–49. doi:10.1016/j.inpa.2016.10.005.

[11] Barbedo JGA, Koenigkan LV, Santos TT. Identifying multiple plant diseases using digital image processing. Biosyst Eng. 2016;147:104–116. doi:10.1016/j.biosystemseng.2016.03.012.

[12] Barbedo JGA. A review on the main challenges in automatic plant disease identification based on visible range images. Biosyst Eng. 2016;144:52–60. doi:10.1109/CVPR.2016.90.

[13] Guo Y, Liu Y, Oerlemans A, et al. Deep learning for visual understanding: a review. Neurocomput. 2016;187:27–48. doi:10.1016/j.neucom.2015.09.116.

[14] Shijie J, Peiyi J, Siping H, et al. Automatic detection of tomato diseases and pests based on leaf images. Proceedings of Chinese Automation Congress; 2017 October 19–22, Ottobre Jinan, China.

[15] Aravind KR, Raja P, Aniirudh R, et al. Grape crop disease classification using transfer learning approach. Proceeding of International Conference on ISMAC in Computational Vision and Bio-Engineering; 2018 May 16–17, Palladam, India, pp. 1623–1633.

[16] Barbedo JGA. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. Comput Electron Agric. 2018a;153:46–53. doi:10.1016/j.compag.2018.08.013.

[17] Ferentinos KP. Deep learning models for plant disease detection and diagnosis. Comput Electron Agric. 2018;145:311–318. doi:10.1016/j.compag.2018.01.009.

[18] Picon A, Gila AA, Seitz M, et al. Deep convolutional neural network for mobile capture device-based crop disease classification in the wild. Comput Electron Agric. in press. doi:10.1016/j.compag.2018.04.002

[19] Too EC, Yujian L, Njuki S, et al. A comparative study of fine-tuning deep learning models for plant disease identification. Comput Electron Agric. in press. doi:10.1016/j.compag.2018.03.0

[20] Hughes DP, Salathe M. An open access repository of images on plant health to enable development of mobile disease diagnostics. arXiv:1511. 2015: 08060.

[21] Lu Y, Yi S, Zheng N, et al. Identification of rice diseases using deep convolutional neural networks. NeuroComput. 2017;267:378–384. doi:10.1016/j.neucom.2017.06.023.

[22] Liu B, Zhang Y, He D, et al. Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry (Basel). 2018;10; doi:10.3390/sym10010011.

[23] Ma J, Du K, Zheng F, et al. A recognition method for cucumber disease diseases using leaf symptom image based on deep convolutional neural network. Comput Electron Agric. 2018;154:18–24. doi:10.1016/j.compag.2018.01.009.

[24] Chen J, Liu Q, Gao L. Visual tea leaf disease recognition using a convolutional neural network model. Symmetry (Basel). 2019;11; doi:10.3390/sym11030343.

[25] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural network. Proceeding of Advances in Neural Information Processing Systems; 2012 December 3–6, Lake Tahoe, USA, pp. 1097–1105.

[26] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Proceedings of International Conference on Learning Representations; 2015 May 7–9, San Diego, USA.

[27] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolution. Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition; 2015 June 7–12, Boston, USA. doi:10.1109/CVPR.2015.7298594.

[28] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition; 2016 June 27–30, Las Vegas, USA, pp. 770-778. doi:10.1109/CVPR.2016.90.

[29] Huang G, Liu Z, Maaten L, et al. Densely connected convolutional network. Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition; 2017 July 21–26, Honolulu, USA, pp. 2261-2269. doi:10.1109/CVPR.2017.243.

[30] Alishiri A, Rakshandehroo F, Zamanizadeh HR, et al. Prevalence of tobacco mosaic virus in Iran and evolutionary analyses of the coat protein gene. Plant Pathol J. 2013;26(3):260–273. doi:10.5423/PPJ.OA.09.2012.0145.

[31] Rao GP, Madhupriya, Thorat V, et al. A century progress of research of phytoplasma disease in India. Phtypathog Mollicutes. 2017;7(1):1–38. doi:10.5958/22494677.2017.00001.9.

[32] Mohanchander P, Reegan AD, Rahman MAU, et al. Effect of hadda beetle Epilachna vigintioctopunchata Fab. (Coleoptera: Coccinellidae) infestation of eggplant leaf (*Solanum melongena* L.) and bio-control potential of essential oil formulations. Pak J Biol Sci. 2016;16(9):991–997. doi:10.3923/pjbs.2013.991.997.

[33] Ramaraju K. Evaluation of acaricides and TNAU neem oils against spider mite, *Tetranychus urticae* (Koch) on bhendi and brinjal. Madras Agric J. 2004;91: 7–12.

[34] Dhakad KK. Epidemiology and management of eggplant diseases in Jabalpur, Madhya Pradesh, Master's thesis, Jawarharlal Nehru Krishi Vishwa Vidyalaya, Jabalpur, India, 2017.

[35] Muedi HTH, Fourie D, Mclaren NW. Characterisation of bacterial brown spot pathogen from dry bean production areas of South Africa. Afr Crop Sci J. 2011;19:357–367.

[36] Navia D, Marasaro AL. First report of the citrus hindu mite, *Schizotetrancyhus hindustanicus* (Hirst) (Prostigmata: Tetranychidae), in Brazil. Neotropic Entomol. 2010;39(1):140–143.

[37] Brungnara EC, Thedoro GFT, Nesi CN, et al. Damage caused by citrus canker on Valencia sweet orange yield. IDESIA(Chile). 2012;30(1):109–113.

[38] Al-Hamadany MN, Al-Karboli HH. First record on okra leafhopper, *Amrasca biguttula biguttula* Ishida on okra in Iraq. Int J Agric Tech. 2017;13(3):393–402.

[39] Kumar A, Verma RB, Kumar R, et al. Yellow vein mosaic disease of okra: A recent management technique. Int J Plant Soil Sci. 2017;19(4):1–8.

[40] Amara J, Bouaziz B, Algergawy A. Deep learning-based approach for banana leaf disease classification. BTW Workshop, 2017, 79–88.

[41] Toda Y, Okura F. How convolutional neural network diagnose plant disease. Plant Phenomics. 2019. Article ID 9237136. doi:10.1155/2019/9237136

[42] Brahimi M, Mahmoudi S, Boukhalfa K. Deep interpretable architecture for plant disease classification. arXiv:1905. 2019: 13523. [cs.CV].

[43] Barbedo JGA. Factors influencing the use of deep learning for plant disease recognition. Biosyst Eng. 2018b;172:84–91. doi:10.1016/j.biosystemseng.2018.05.013.