Hindawi
Complexity
Volume 2022, Article ID 1168700, 11 pages
https://doi.org/10.1155/2022/1168700

Research Article

End-to-End Semantic Leaf Segmentation Framework for Plants Disease Classification

Khalil Khan,[1,2] **Rehan Ullah Khan**,[3] **Waleed Albattah**,[3] **and Ali Mustafa Qamar**[4]

[1]*Department of Information Technology and Computer Science, Pak-Austria Fachhochschule,*
Institute of Applied Sciences and Technology, Haripur 22620, Pakistan
[2]*Faculty of Computer Sciences and Information Technology, Superior University, Lahore 54660, Pakistan*
[3]*Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia*
[4]*Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia*

Correspondence should be addressed to Rehan Ullah Khan; re.khan@qu.edu.sa

Received 4 March 2022; Revised 14 April 2022; Accepted 9 May 2022; Published 30 May 2022

Academic Editor: Shahzad Sarfraz

Pernicious insects and plant diseases threaten the food science and agriculture sector. Therefore, diagnosis and detection of such diseases are essential. Plant disease detection and classification is a much-developed research area due to enormous development in machine learning (ML). Over the last ten years, computer vision researchers proposed different algorithms for plant disease identification using ML. This paper proposes an end-to-end semantic leaf segmentation model for plant disease identification. Our model uses a deep convolutional neural network based on semantic segmentation (SS). The proposed algorithm highlights diseased and healthy parts and allows the classification of ten different diseases affecting a specific plant leaf. The model successfully highlights the foreground (leaf) and background (nonleaf) regions through SS, identifying regions as healthy and diseased parts. As the semantic label is provided by the proposed method for each pixel, the information about how much area of a specific leaf is affected due to a disease is also estimated. We use tomato plant leaves as a test case in our work. We test the proposed CNN-based model on the publicly available database, PlantVillage. Along with PlantVillage, we also collected a dataset of twenty thousand images and tested our framework on it. Our proposed model obtained an average accuracy of 97.6%, which shows substantial improvement in performance on the same dataset compared to previous results.

1. Introduction

The plants' diseases in crops and fruits have adverse effects on agriculture production. If these diseases are not identified and treated on time, an increase in food insecurity can occur. Some particular crops, such as wheat, rice, and maize are vital for ensuring the food supply as well as agriculture production. Early warnings and some forecasting are very effective prevention in controlling plants' diseases. Forecasting and prevention play an essential role in adequately managing agricultural production. However, until now, visual observations of producers are the only approach for various plants' disease identification in mostly rural areas, specifically in less developed countries. Continuous monitoring of experts is needed, which might be prohibitively very expensive in large farms.

Similarly, to contact experts' farmers may have to travel large distances, which also makes the consultation expensive and time consuming. We argue that this conventional approach is not practical for large farming areas looking into the demands of the crops in the production industry. Therefore, automatic plant disease recognition and classification are still crucial topics in computer vision (CV).

Diseases seriously affect the health of every living organism, including plants and animals. The state-of-the-art (S.O.A) algorithms in the CV and ML domains have enabled us to identify diseases beyond human accuracy. Interestingly, the CV and ML algorithms can be applied to all domains, including plants and humans, with almost no difference in implementation. Modern technology enables human society to generate sufficient food to fulfill the human population's requirements. Conversely, numerous factors still affect food safety, including plant diseases, climate change, and the decline of pollinators. Plant diseases are the primary danger to food safety. At the same time, the deficiency of essential infrastructure makes it hard to identify these diseases in many parts of the world quickly. With the latest developments in CV algorithms, the ML paradigm paves its way for agile and on-spot disease diagnosis.

Plant diseases not taken seriously have caused a decline in agricultural productivity in several countries worldwide. The disease symptoms have a detrimental effect on crop growth, restraining yields and rendering agricultural goods unsuitable to be consumed. Therefore, their early detection, modeling, and recognition are essential. This article explores the modeling, detection, and recognition of plant diseases that involve appearance-based analysis and can be captured and modeled using ML. Since the leaves of plants provide expressive appearance-based modeling, from modeling perspectives, our interest is inclined towards disease detection in tomato plants using deep learning (DL).

Determining the health quality of a plant is essential. Several models have been created to deter the loss of crops to pests and diseases. Plant disease symptoms are usually noticeable when the leaves change color or shape. Traditionally, the identification of pests and diseases was done using the naked eye and was supported by agronomic organizations. Presently, detection of plant diseases and pests can be done through machine vision. Plant disease identification using ML is not a new research field. CV experts have reported many good papers worth mentioning in [1–6]. The extensive penetration of smartphones, high-density cameras, and high-performance processors have made it possible for diseases to be detected by automated image recognition.

This paper proposes an end-to-end (E2E) segmentation model for plant disease identification and classification. The model uses semantic leaf segmentation (SLS) using an optimized CNN. By successfully highlighting the foreground and background regions, the proposed model classifies them into healthy and disease parts. The proposed model encodes the high-density maps and classifies tomato plant leaves into ten different categories of various diseases. Our model outperforms previous approaches in an evaluation setup on the PlantVillage database. The significant contributions of the proposed work are as follows:

(i) A new CNN-based algorithm for plants' disease recognition and classification has been presented in the paper. ML and DL experts have already proposed numerous methods for plants disease recognition; the novelty of our model is providing information about each pixel of a leaf image, which tells if a pixel belongs to a diseased or healthy part.

(ii) Second, we contributed a new dataset for tomato leaves disease classification. We collected these images from the Internet. These images are labeled manually as healthy and diseased images. The database will be available after the publication of the proposed work for research purposes.

(iii) The proposed model also provides information regarding how much leaf area is affected by a specific disease. Most of the previous ML-based approaches do not provide this information. Along with predicting the diseased and healthy part, information regarding how much leaf area is affected by a specific disease is also provided.

The presentation of the remaining paper is arranged as follows. In Section 2, we discuss previous research work on the topic. Both conventional ML and DL-based methods are discussed in this part. The proposed CNN-based method is discussed in Section 3. Section 4 presents the experimental setup, obtained results, and comparison of obtained results with previous results. Finally, the conclusion is presented in Section 5 with some promising future directions.

2. Related Work

Plant diseases typically affect the growth of crops in all stages of development and sometimes may lead to the death of the plant. Plant diseases affect food security globally and affect small subsistence farmers who depend on their crops for food and livelihood. Therefore, determining the health quality of a plant is very important. Several models have been created to deter the loss of crops to pests and diseases. Traditionally, the identification of pests and diseases was done using the naked eye and was supported by agronomic organizations. Plant disease identification using ML is a well-researched area. CV experts have reported many good papers on the topic [1–9].

In meticulous agriculture, the subdivision of crops in agricultural images is vital. Various techniques have been deployed for the segmentation process, such as SS. The SS marks the multiple features in an image into semantically meaningful items and classifies each item into a class. For example, the various classes can be leaf, stalk, or flower in plants. Several studies have used different SS techniques to identify plants from nonplants. For example, Sodjinou et al. [10] suggest a method grounded on the mixture of SS and K-means for detecting weed from images. K-means algorithm is used for categorizing things that belong to similar groups. The proposed technique provided a more accurate segmentation of weeds and plants from the study results. Several approaches were used by Miao et al. [11] to semantically segment hyperspectral images of sorghum plants, such as manual pixel annotation and classifying each of the pixels as either nonplant or plant. The scholars further classified the plant as belonging to either a panicle, leaf, or stalk of the sorghum plant. They could separate the plant pixels from the background, only that they could not classify to what organ the plant pixel belonged. In another study, Li et al. [12] used the region-based segmentation to detect crops from the images derived from a natural field. They used the method to detect cotton specifically. The model was successful as it could even detect the boll opening stage of the cotton plant.

While identifying a plant from a nonplant can be an easy task using SS, identifying plant diseases through an image is tough since plants are complex environments. Through the developmental stages of crops, their flowers, fruits, and leaves change constantly. During the day, solar radiation affects plants' spectral response, so their appearance also changes slightly. Additionally, different shapes, layouts, and colors of plant diseases make them difficult to recognize. Regardless, several successful techniques improve detection methods for diseases in plants, both in a controlled environment and in natural conditions.

Chen et al. [13] proposed using BLSNet to recognize the rice bacterial leaf streak in rice and segmentation based on UNet network. Rice bacterial leaf streak (BLS) is a threatening disease usually found in rice leaves. BLS affects the yield and quality of rice. BLSNet used a large-scale extraction and an attention mechanism to increase the precision of segmentation of the lesion.

One technique that has been widely successful in identifying plant diseases is SS through CNN. The layers of CNN can be viewed as corresponding filters that are directly taken from the input data. CNNs bring out a hierarchy of visual images adjusted for a precise task. The accuracy of CNNs in detecting objects such as plant diseases and image organization has made incredible growth over time [14]. The CNN-based classification network implementation is the most regularly used pattern in categorizing plant diseases and pests, owing to CNN's strong feature extraction capability. Zabawa et al. [15] used SS using convolutional neural networks to extract phenotypic traits in grapevine berries.

According to Bhatt et al. [16], CNN-based methods have been used to achieve extraordinary results in supervised image segmentation of leaves. Usually, the methods used work under fully controlled conditions, whereas the deep CNN models are built on various changing parameters. However, the images would have different backgrounds, lighting conditions, obstructions, and overlapping in an environmental setting. In various stages of growth, plants do have a reasonable amount of variation. The authors propose unsupervised machine learning algorithms to segment the leaves images to make it possible to be applied to various crops and regions. Afterward, the specific segments are then assessed for their texture, size, and color to measure any change, such as the presence of a pest or disease.

Unsupervised feature learning, with fully convolutional networks (FCN) followed by conditional random fields, makes it possible to segment images into an optimal number of clusters devoid of any prior training. The real-time performance of this technique allows easy distribution of devices such as cameras and mobile phones in the fields. In addition, Shao et al. [17] propose using localization and DL-based method to recognize dense rice images. The proposed model can be used to determine rice diseases. The results from the study show that better results can be obtained compared to conventional ML methods. The SS method grounded on deep CNNs can also identify crops from the compound and natural field environments [18]. According to Martins et al. [19], it can also detect tree canopies in an urban setting.

The SS method based on DL demonstrates great precision in remote sensing categorization as well, and it necessitates vast sets of data in controlled learning [20]. The simple notion of DL is using a neural network for analyzing information and learning image feature. In their study of estimating sorghum panicles, Malambo et al. [21] applied an image analysis method founded on a SegNet framework. Sorghum panicles are critical phenotypic data in the improvement of sorghum crops. The study results demonstrated that DL combined with SS shows excellent precision with large data. On the other hand, Pena et al. [22] suggest using data fusion to enrich images used in remote sensing.

In very recent works [23–26], plant disease recognition models have been improved for better results and performance. Manjula et al. [24] have used ResNet-50 architecture, a variant of the Resnet model that has 48 convolution layers. The accuracy of the developed system is around 97-98%. Chen et al. [25] have improved the plant disease-recognition model based on the original YOLOv5 network model, which accurately identified plant diseases under natural conditions. Hassan and Maji [26] have proposed a novel deep learning model based on the inception layer and residual connection. They used Depthwise separable convolution to reduce the number of parameters, which led the model to achieve higher accuracy.

3. Materials and Methods

3.1. Dataset Description and Data Annotation. The typical DL-based methods require sufficient data for the training phase. In contrast, conventional ML can also be trained on limited data scenarios. One of the main drawbacks of DL techniques is requiring a large amount of data. In this research work, we used an already available dataset and also collected our own database. We use two kinds of data in our experiments, the details of which are provided next.

PlantVillage database [27]: The PlantVillage database is publicly available for downloading and research purposes. It is an open-access repository having more than 54K images. PlantVillage is a large dataset with various plants' leaves and related materials collection. Most of the data in this database are collected in controlled laboratory conditions. Exposure to the real-time scenario is significantly less in the PlantVillage database. Therefore, most researchers using only PlantVillage database get nearly perfect results. The database includes images of 14 crops, including grape, corn, tomato, and soybean. The database consists of 10 folders, one for healthy leaves and the remaining for nine different kinds of diseases listed in Tables 1 and 2. We use a subset of images for the tomato plant. Our subset consists of around 16012 images of plant leaves collected from tomato plants. The total number of classes in these images is limited to ten only. Nine classes are of various diseases for tomato plant leaves, whereas one class is for healthy leaves. We keep the resolution of each image as *250 x 250* pixels. Some sample images of the database used are shown in Figure 1.

Table 1: Number-wise distribution of the PlantVillage database.

Disease	Number of images
Bacterial spot (BS)	2127
Late blight (LB)	1909
Two-spotted spider mite (TSSM)	1676
Yellow leaf curl virus (YLCV)	5357
Leaf mold (LM)	952
Target spot (TS)	1404
Early blight (EB)	1000
Tomato mosaic virus (TMV)	373
Septoria leaf spot (SLS)	1771

Table 2: TomatoDB images and different diseases' statistics.

Disease	Number of images in TomatoDB
EB	2250
LB	1800
TMV	2190
HL	1600
LM	2000
YLCV	1802
BS	2380
SLS	2100
TS	2270
SM	2340
Total	20732

We use all images of the tomato plant contained in the PlantVillage. The diseased leaf images vary from 373 to 5357, as clear from Table 1. The total number of healthy images for tomatoes in PlantVillage is 1591. It is clear from Table 1 that all the ten classes in the dataset are not balanced as far as the number of images is concerned. On the one hand, the minimum value is 373, with a maximum of 5357. We use data augmentation methods to balance all classes, including adjusting the contrast, flipping images vertically and horizontally, and changing brightness levels.

TomatoDB: since images in the PlantVillage are simple and less challenging, comparatively good results are reported in the literature. To assess the framework's performance more precisely, we also tested our model on a collection of images we had taken from the Internet. Our own collected dataset consists of more than 20000 images taken from tomato plant leaves. We collected these images from the Internet. While image collection, real-time scenarios, and more challenging conditions have been considered. The database TomatoDB will be available to the research community after the publication of the proposed research article. All ten classes are equally considered while collecting the database.

For SLS, correctly labeled leaf data for each pixel is needed. This ground truth data are created through annotation. We annotated these images manually using the interface we developed. This labeling involves selecting the areas of interest, random sketch application, adjustment of contrast and brightness, and assigning a label. Such kind of manual labeling is prone to errors. No automatic tool is used in such labeling. The labeling is highly dependent on the subjective perception of the human doing this labeling process. Hence, chances of error exist while providing an exact label to every pixel.

Images setup for experiments: The model we presented in this paper is applicable and valid to any plant disease with some visible symptoms. However, manual labeling will be needed to create an SS framework for training purposes. As a test case, we select a tomato plant with ten classes. However, since images in the PlantVillage dataset are not exposed to light and other variations, we collected some images (20000) from the Internet. Some tomato leaves images we collected from the Internet are shown in Figure 2. We use a combination of PlantVillage dataset images and our own collected data set. We split the dataset into the ratio of 80 to 20, a commonly used strategy for training and testing DL-based models. Some of the authors also adopt 5-fold or 10-fold cross validation. We set 80% data for training and 10% for validation to know the model overfitting problem. We resize each image in PlantVillage and TomatoDB to a size of *250 x250* before training and testing. The following section discusses all the hyperparameters of the deep CNN-based model.

3.2. Deep Model Learning. The performance of the visual recognition tasks is improved with the introduction of DL-based methods [28–33]. The proposed paper addresses leaf disease recognition and classification using deep CNNs. We utilize the concept of SLS in the proposed research.

Convolution layer: This layer plays a vital role in the features extraction stage. The CovL is an essential component of the CNN model. The layers consist of a set of learnable filters. These terms are also known as kernels [34]. In this convolution process, the filter with a specific size slides over the image and is convolved with pixel values of the target image. The dot product is computed between kernel and input image pixels producing a feature map.

ReLU: We use ReLU as activation function. This function plays a crucial role in converting the input signal from a specific network node to the output signal. The resultant signal obtained a form as shown in equation.

$$f(F) = \max\left(0, F_j\right). \tag{1}$$

Pooling layer: The pooling layer follows the CovL. The output from the CovL is given to the pooling layer. ML experts use three pooling strategies: random pooling, maximum pooling (MPL), and average pooling. We, in the proposed work, adapted MPL. The MPL achieved spatial invariance by reducing the feature map size obtained previously from CovL [35]. In this strategy, the max operation is applied to the feature map when the feature map is passed through MPL. This operation can be performed as described by :

$$\mathrm{MPL}_j = \max\left(F_i\right). \tag{2}$$

Classification: we use the SoftMax classifier for classification. The pooling layers provide a feature vector to the

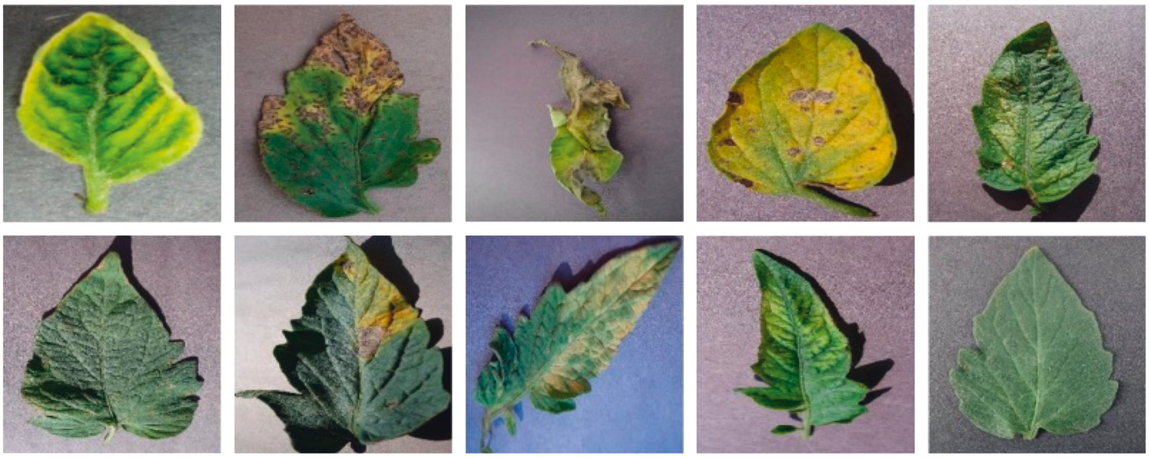

Figure 1: Sample images from PlantVillage database.

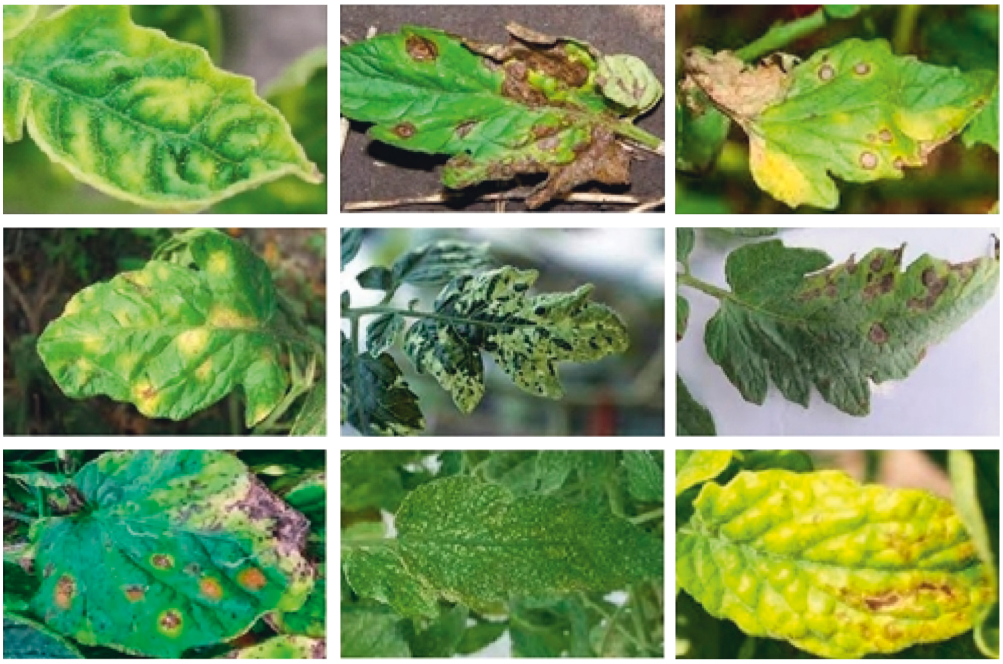

Figure 2: Some images of the leaves from the TomatoDB dataset.

SoftMax in the output layer. In the output layer, a function that appears is the activation function for a multiclass classification problem. The activation function calculates a vector having a real number (k) and performs the normalization task. The normalization converts input values into vectors consisting of probability values in the range 0 to 1. The Softmax returns each class probability value, having the maximum probability value as the target class [36].

Adam: It is a standard optimizer that computes individual adaptive learning rates for each parameter [37]. The exponential decaying average of previous gradients n_t is used by this optimizer.

The proposed framework is presented in Figure 3.

Tables 3 and 4 summarize various parameters in the proposed CNN framework. As an activation function, we use ReLU. For constructing CNN-based model, we use three layers containing CovL, MPL, and FCL. The details of these layers with feature map description, kernel size, and stride are summarized in Table 3. The feature extractor extracts the features from the images of the leaves, including healthy and affected leaves. More description of the feature extraction part is in Figure 4. Features variation is handled by stage 1. Certain environmental factors produce scaling variations in images. These receptive fields overcome all the variations. Each field has sixteen filters. Stage 1 output is given to stage

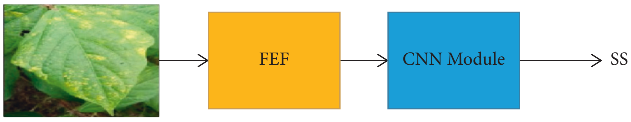

Figure 3: SLS-based leaf disease identification model.

Table 3: CovL and MPL parameters' setting.

Type of layer	Size of stride	Feature map	Size of kernel	Output size
Input	—	—	—	250 × 250
CovL1	2	96	5 × 5	124 × 124
MPL1	2	96	3 × 3	62 × 62
CovL2	2	256	5 × 5	30 × 30
MPL2	2	256	3 × 3	15 × 15
CovL3	2	316	5 × 5	12 × 12
MPL3	2	316	3 × 3	6 × 6
CovL4	2	512	5 × 5	4 × 4
MPL4	2	512	3 × 3	2 × 2

Table 4: Deep CNN parameters' settings.

Parameters	Values
Number of epochs	500
Activation function	ReLU
Momentum	0.99
Batch size	150
Base learning rate	0.0001
Drop our rate	0.40

2. We use the 2 × 2 kernel in MPL in stage 2. Each layer of ConvL is followed by ReLU. We place a special pyramid (SPD) between CovL and FCL. In stage 3, output from SPD is given to FCL. Both stages 3 and 4 extract desired features. More details about deep CNN parameters are presented in Tables 3 and 4 and Figure 4.

Data including both training images and ground truth are given to the framework. The density map is predicted in density estimation (DE), taking supervision from the ground truth data. We combined the segmentation map and DE map, feeding the results to the CovL. Loss is added to the algorithm (Dice Coefficient) in the SS section. Additionally, we add Euclidean distance loss for optimizing the estimated density maps.

3.3. CNN Optimization. A complete illustration of how hyperparameters are tuned and optimization is performed is presented in this subsection. Overfitting is a severe problem faced mainly by ML models. We use the methodology as suggested and used in [38] to tackle this problem. We use a combination of four different loss functions. We use the Euclidean distance for better optimization. The obtained segmentation density map can be written as shown in (3) and (4)

$$\mathrm{Loss}_{\mathrm{int}} = \frac{1}{2M}\sum_{k=1}^{M} \widehat{p}_k - P_{k2}^{\;2}. \tag{3}$$

$$\mathrm{Loss}_{den} = \frac{1}{2M}\sum_{k=1}^{M} \widehat{P}_k - P_{k2}^{\;2}. \tag{4}$$

In (3), $\widehat{p}$ shows the estimated density in the supervision process. Similarly, $\widehat{P}_k$ represents the estimated density, and P_k represents the ground truth density value. Similarly, M represents the pixel numbers in the GT density map.

We also introduce a loss in the SS part of the framework. The loss in the framework is due to the dice coefficient. The dice coefficient is two times the overlap area between the predicted segmentation and true values. The result is then divided by the total pixels in the ground truth and the original image. The range of the dice coefficient is between 0 and 1. We use another special loss function, called cross-entropy loss, which we represent as

$$\mathrm{Loss}_{X-\mathrm{entropy}} = -\frac{1}{Q}\sum_{b=1}^{Q}\sum_{c=1}^{C} x_c^b \log\left(x^b\right). \tag{5}$$

In (5), the symbol Q represents the total sample, and C shows the number of classes used. Similarly, the ground truth class is shown by x_c^b, whereas the estimated output is represented by x^b. The final weighted loss function is represented by :

$$W.L = \mathrm{Loss}_{\mathrm{int}} + \mathrm{Loss}_{den} + \lambda \mathrm{Loss}_{X-\mathrm{entropy}}. \tag{6}$$

In (6), the value of λ was 0.3.

4. Results and Discussion

4.1. Performance Evaluation Measures. We use different evaluation measures, including precision (P_r), recall (R_c), accuracy (A_{cc}), F-measure (F_m), and confusion matrix (C_{mat}). Most of these measures are defined with some terms called false positive (FP), true positive (TP), true negative (TN), and false-negative (FN). P_r is lower if the number of FPs is more. The R_c measures the correct prediction (positive only) by calculating the proportion of the number of TPs to the total sample (TP + FN). The range of both P_r and R_c is between 0 and 1. The F_m assesses the performance of the model by calculating the weighted harmonic mean between P_r and R_c. Mathematically, all the evaluation measures are defined in equations (7)–(10):

$$P_r = \frac{(\mathrm{TP})}{(\mathrm{TP}) + (\mathrm{FP})}. \tag{7}$$

$$R_c = \frac{(\mathrm{TP})}{(\mathrm{TP}) + (\mathrm{FN})}. \tag{8}$$

$$F_m = 2 * \frac{(\mathrm{Precision})x(\mathrm{Recall})}{(\mathrm{Precision}) + (\mathrm{Recall})}. \tag{9}$$

$$A_{cc} = \frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})}. \tag{10}$$

4.2. Experimental Setup. We perform our experiments with an Intel i7 workstation and employ NVIDIA GPU 840 graphics card. We perform all our experimental work with Tensor-flow, Keras, and *Python*. The number of epochs we use is 500, having a batch size of 150. We use the base learning rate as 0.0001 and the dropout rate as 0.4. We use two datasets for experimental work: including PlantVillage

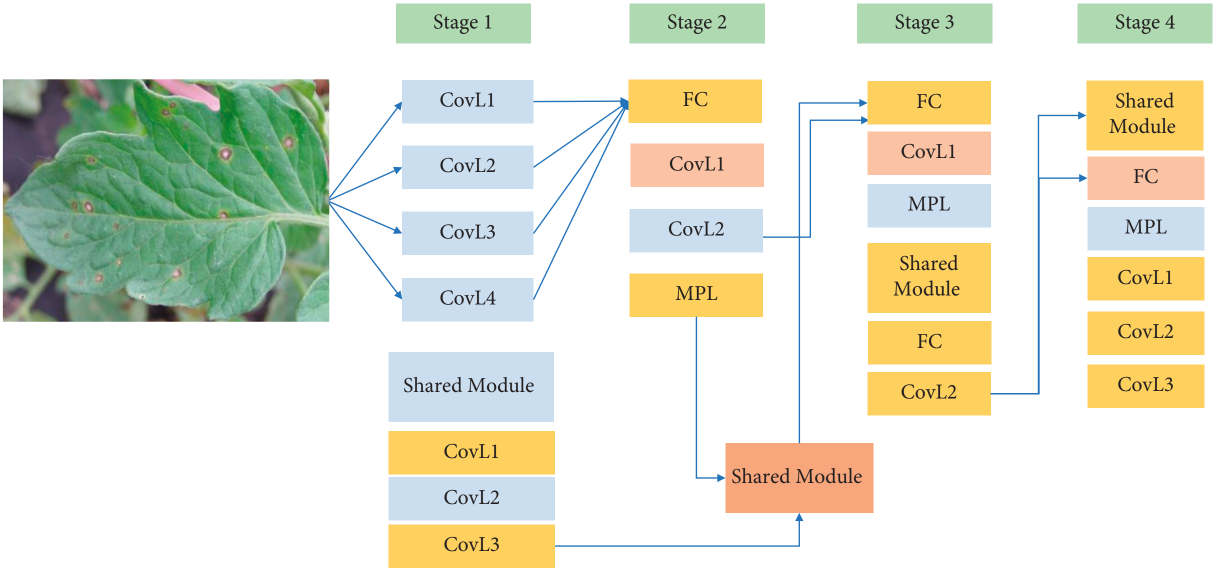

Figure 4: Proposed feature extractor module.

and our own collected database. The PlantVillage consists of more than 16K images, and our own database consists of more than 20K images. We combined both datasets and performed our experiments in the ratio of training to testing as 80 to 20.

4.3. Limitations of the Proposed Work. The numerical solutions and results reported in this paper show that a good performance is achieved by the proposed method; however, our proposed algorithm still has limitations. It is a fact that the research community has concerns about using DL architectures. All DL-based methods are complex and require inputs at several stages. Researchers using these techniques rely on a trial and error strategy. To summarize, these methods are time consuming and very well engineered. However, it is also confirmed that the only choice CV experts have for any CV-based task is DL methods. We use the idea of SLS in our proposed work. Ground truth data are needed for the training and testing phases to implement this model. In order to create the ground truth data, manual labeling is required. Since a single person does all this manual labeling, errors are expected most of the time in labeling. We also did this labeling manually through humans, which is a weakness of our proposed method.

4.4. Reported Results and Its Discussion. Some conclusions that emerge from the results and experiments are summarized in the following paragraphs.

(i) Plant disease classification and identification using ML is not a new research area for CV and ML experts. The state-of-the-art reports many good papers on this topic. Due to diverse applications in agriculture, researchers explored this field sufficiently. However, we notice less emphasis, particularly on interclass disease identification. Researchers mainly focus on a single plant disease recognition, whereas our proposed work focuses on tomato plant disease classification with ten classes.

(ii) Initially, we run the whole experimental setup for a maximum of 14 epochs (please see Figure 5). We run this setup to know how the model performance varies on training and validation databases. As clear from Figure 5, training along with validation accuracy changes very quickly up to value 6. After value 6, change occurs very slowly in the upcoming epochs. Both training and validation losses are also shown. It is clear that loss is high in the initial stages and is gradually reduced after increasing the epochs. This loss reduction clearly shows that the network is fine-tuned gradually with increasing epochs.

(iii) We use ten class disease problems in our work. The names of the classes, along with abbreviations, are shown in Table 5. We report the results for P_r, R_c, and F_m for all the ten classes. It is clear from Table 5 that near-perfect results are reported for the class BS using all three evaluation measures. Similarly, better results are reported for the classes LB, TMV, SM, and HL. The worst performance has been shown for the class EB with precision 0.93, recall 0.95, and F-measure 0.95, which also shows acceptable and good results. Our proposed method semantically segments leaf images into background and foreground. Please see some images in Figure 6, where column 1 represents the original images, column 2 ground truth, and column 3 segmentation results. After foreground estimation, each disease classification is performed. Moreover, it is also estimated how much percent of the leaf area is affected by a disease.

(iv) C_{mat} is the best choice for multiclass evaluation problems, which ML experts commonly use. It shows the corresponding percentage of the

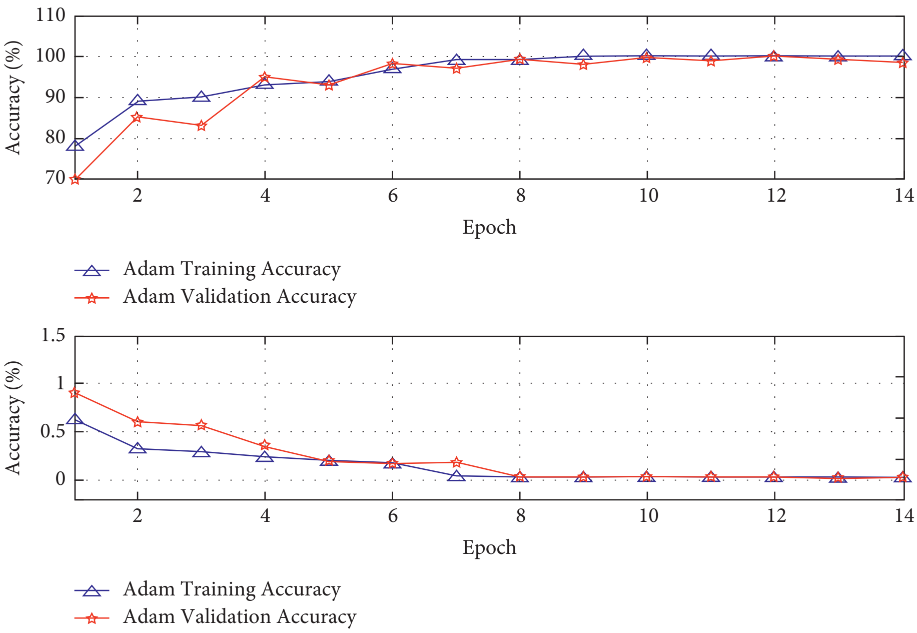

FIGURE 5: Using Adam optimizer, accuracy, and loss changing obtained using training and validation sets.

TABLE 5: Performance reported in the form of P_r, R_c, and F_m for the proposed model.

Class	Precision	Recall	F-measure
EB	0.93	0.95	0.95
LB	0.96	0.99	1.00
TMV	0.97	0.99	0.96
HL	0.96	0.97	0.97
LM	0.93	0.95	0.98
YLCV	0.94	0.93	0.96
BS	0.96	0.99	1.00
SLS	0.96	0.96	0.98
TS	0.96	0.94	0.94
SM	0.96	0.98	0.99
Mean	0.95	0.96	0.97

predicted class and true class. The C_{mat} for the reported results for the 10-class problem is demonstrated in Table 6. The results vary from 94% (lowest) to 100% (highest). The lowest results are reported for EB, whereas the highest values are reported for HL. The LM, YLCV, BS, and LB results are comparatively better, with predicted accuracy values as 99%, 98%, 98%, and 97%, respectively.

4.5. Performance Comparison with Previous Results. We compared the reported results with S.O.A. in Table 7. It is clear that the reported results are far better than previous results. The reported results and their comparison with S.O.A. are for accuracy measure only. As most of the papers reported their accuracy results, we compared our work with this metric only. We want to add that some research papers reporting results on plant disease classification using handcrafted features show better results than DL-based methods. However, we believe a better understanding of DL methods

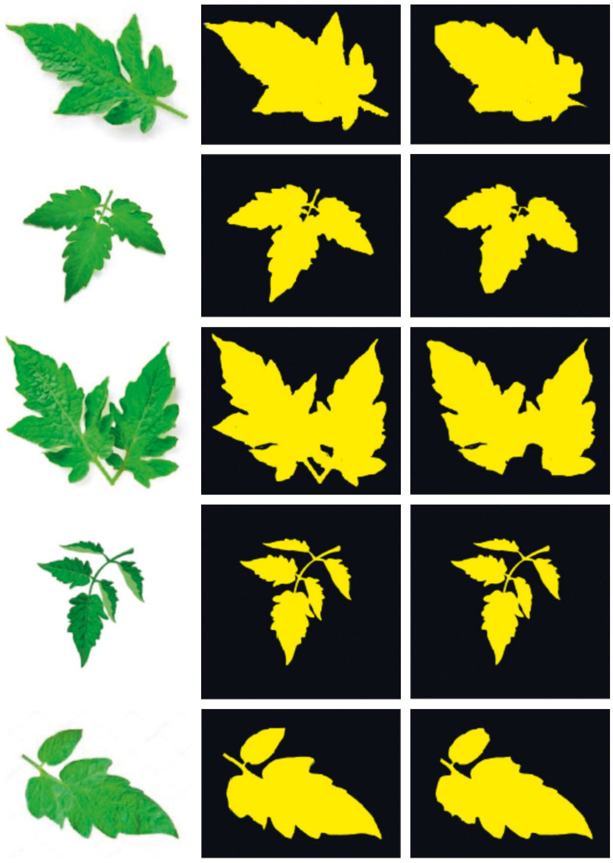

FIGURE 6: Some example images we used during our experiments: column 1 shows original images, column 2 shows ground truth data, and column 3 shows segmentation images with the proposed SLS method.

Table 6: Reported confusion matrix for tomato plant diseases recognition with proposed SLS-based method.

		Predicted class									
		EB	*LB*	*TMV*	*HL*	*LM*	*YLCV*	*BS*	*SLS*	*TS*	*SM*
True class	*EB*	**94**	1	0	0	2	1	1	1	0	0
	LB	1	**98**	0	0	0	0	0	0	1	0
	TMV	0	0	**100**	0	0	0	0	0	0	0
	HL	0	0	0	**100**	0	0	0.24	0	0	0
	LM	0	0	0	0	**99**	1	0	0	0	0
	YLCV	0	0	0	0	0	**98**	0	1	1	0
	BS	1	0	0	0	0	0	**98**	1	0	0
	SLS	0	0	0	0	0	1	1	**96**	1	1
	TS	0	0	0	0	0	2	1	1	**95**	1
	SM	0	0	0	0	0	0	1	1	1	**97**

Table 7: Reported results and their comparison with previous results.

Method used	Year	Algorithm (s)	Reported accuracy (%)
Proposed method	2022	SLS and CNNs	97.25
Abbas et al. [39]	2021	DenseNet	95.1
Agarwal et al. [40]	2020	GNN-based network	91.2
Elhassouny and Smarandache [41]	2019	MobileNet	90.3
Durmuş et al. [42]	2017	Combination of squeez net and AlexNet	95.65
Howard et al. [43]	2017	MobileNet	63.7
Szegedy et al. [44]	2016	Inception V3	63.4
Simonyan and Zisserman [45]	2014	VGG 16	77.2

is still required to address a specific task. For example, the requirement of a large amount of data is a problem DL methods face. Generally, traditional ML methods perform well on data collected in indoor scenes; however, researchers report a significant drop in performance when these methods are tested in real-time scenarios. On the contrary, DL architectures extract a higher level of abstraction from the data with much better results. Thus, the need for feature engineering is minimized to a large extent with DL algorithms.

5. Summary and Concluding Remarks

Due to diverse applications in the agriculture sector, plant disease identification using DL is an active area of research. Plant disease recognition is more challenging when the method is exposed to real-time data. However, CV researchers have shown tremendous progress in the past 5 to 10 years. Our current research provides unification and extension of our previous work reported in [7]. Our study is mainly motivated by looking into the human visual cortex to design an E2E trainable neural network architecture. We propose an E2E SS framework for plant disease identification using DL. We introduce the idea of SS for plant disease recognition. The proposed model predicts the nature of the disease of the tomato plant and tells how much area of a specific leaf is affected due to a certain disease. The model successfully classifies tomato plant leaves into ten distinct classes. We present a novel loss function that improves the model's performance on a state-of-the-art dataset. We evaluate our model with the standard dataset PlantVillage, noticing much better results than previous results. Along with the PlantVillage database, we also collected a database of more than 20000 images and tested our framework on it. We expect more evaluation using a much better optimized DL model for plant disease recognition from the research community. In the future, we intend to analyze some more tasks to develop robust continual DL models, considering some complex combinations of the neural network along with information extraction.

Data Availability

The dataset used in this research, PlantsVillage, is available from https://www.kaggle.com/emmarex/plantdisease.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors gratefully acknowledge Qassim University, represented by the Deanship of Scientific Research, on the financial support for this research, under the no. 10269-coc-2020-1-3-I, during the academic year 1441 AH/2020 AD.

References

[1] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019.

[2] J. G. Arnal Barbedo, "Plant disease identification from individual lesions and spots using deep learning," *Biosystems Engineering*, vol. 180, pp. 96–107, 2019.

[3] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Computers and Electronics in Agriculture*, vol. 173, Article ID 105393, 2020.

[4] M. H. Saleem, J. Potgieter, and K. M. Mahmood Arif, "Plant disease detection and classification by deep learning," *Plants*, vol. 8, no. 11, p. 468, 2019.

[5] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.

[6] M. H. Saleem, S. Khanchi, J. Potgieter, and K. M. Arif, "Image-based plant disease identification by deep learning meta-architectures," *Plants*, vol. 9, no. 11, p. 1451, 2020.

[7] R. U. Khan, K. Khan, W. Albattah, and A. M. Qamar, "Image-based detection of plant diseases: from classical machine learning to deep learning journey," *Wireless Communications and Mobile Computing*, vol. 202113 pages, Article ID 5541859, 2021.

[8] D. Srinivasa Rao, R. Babu Ch, V. Sravan Kiran et al., "Plant disease classification using deep bilinear CNN," *Intelligent Automation & Soft Computing*, vol. 31, no. 1, pp. 161–176, 2022.

[9] S. Nandhini and K. Ashokkumar, "An automatic plant leaf disease identification using DenseNet-121 architecture with a mutation-based henry gas solubility optimization algorithm," *Neural Computing & Applications*, vol. 34, no. 7, pp. 5513–5534, 2022.

[10] S. G. Sodjinou, V. Mohammadi, A. T. S. Mahama, and P. Gouton, "A deep semantic segmentation-based algorithm to segment crops and weeds in agronomic color images," *Information Processing in Agriculture*, 2021.

[11] C. Miao, A. Pages, Z. Xu, E. Rodene, J. Yang, and J. C. Schnable, "Semantic segmentation of Sorghum using hyperspectral data identifies genetic associations," *Plant Phenomics*, vol. 202011 pages, Article ID 4216373, 2020.

[12] Y. Li, Z. Cao, H. Lu, Y. Xiao, Y. Zhu, and A. B. Cremers, "In-field cotton detection via region-based semantic image segmentation," *Computers and Electronics in Agriculture*, vol. 127, pp. 475–486, 2016.

[13] S. Chen, K. Zhang, Y. Zhao et al., "An approach for rice bacterial leaf streak disease segmentation and disease severity estimation," *Agriculture*, vol. 11, no. 5, p. 420, 2021.

[14] S. Guo, Q. Jin, H. Wang, X. Wang, Y. Wang, and S. Xiang, "Learnable gated convolutional neural network for semantic segmentation in remote-sensing images," *Remote Sensing*, vol. 11, no. 16, p. 1922, 2019.

[15] L. Zabawa, A. Kicherer, L. Klingbeil, R. Töpfer, H. Kuhlmann, and R. Roscher, "Counting of grapevine berries in images via semantic segmentation using convolutional neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 164, pp. 73–83, 2020.

[16] P. Bhatt, S. Sarangi, and S. Pappula, "Unsupervised image segmentation using convolutional neural networks for automated crop monitoring," in *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pp. 887–893, Prague, Czech Republic, February 2019.

[17] H. Shao, R. Tang, Y. Lei, J. Mu, Y. Guan, and Y. Xiang, "Rice ear counting based on image segmentation and establishment of a dataset," *Plants*, vol. 10, no. 8, p. 1625, 2021.

[18] S. Zhuang, P. Wang, and B. Jiang, "Segmentation of green vegetation in the field using deep neural networks," in *Proceedings of the 2018 13th World Congress on Intelligent Control and Automation*, Changsha, China, July 2018.

[19] J. A. C. Martins, K. Nogueira, L. P. Osco et al., "Semantic segmentation of tree-canopy in urban environment with pixel-wise deep learning," *Remote Sensing*, vol. 13, no. 16, p. 3054, 2021.

[20] M.-D. Yang, H.-H. Tseng, Y.-C. Hsu, and H. P. Tsai, "Semantic segmentation using deep learning with vegetation indices for rice lodging identification in multi-date UAV visible images," *Remote Sensing*, vol. 12, no. 4, p. 633, 2020.

[21] L. Malambo, S. Popescu, N.-W. Ku, W. Rooney, T. Zhou, and S. Moore, "A deep learning semantic segmentation-based approach for field-level sorghum panicle counting," *Remote Sensing*, vol. 11, no. 24, p. 2939, 2019.

[22] J. Pena, Y. Tan, and W. Boonpook, "Semantic segmentation based remote sensing data fusion on crops detection," *Journal of Computer and Communications*, vol. 07, no. 07, pp. 53–64, 2019.

[23] T. S. Poornappriya and R. Gopinath, "Rice plant disease identification using artificial intelligence approaches," *International Journal of Electrical Engineering and Technology*, vol. 11, no. 10, pp. 392–402, 2022.

[24] K. Manjula, S. Spoorthi, R. Yashaswini, and D. Sharma, "Plant disease detection using deep learning," in *Proceedings of the 2nd International Conference on Data Science, Machine Learning and Applications (ICDSMLA)*, pp. 1389–1396, Singapore, April 2020.

[25] Z. Chen, R. Wu, Y. Lin et al., "Plant disease recognition model based on improved YOLOv5," *Agronomy*, vol. 12, no. 2, p. 365, 2022.

[26] S. M. Hassan and A. K. Maji, "Plant disease identification using a novel convolutional neural network," *IEEE Access*, vol. 10, pp. 5390–5401, 2022.

[27] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," 2015, https://arxiv.org/abs/1511.08060.

[28] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: a brief review," *Computational Intelligence and Neuroscience*, vol. 201813 pages, Article ID 7068349, 2018.

[29] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: a survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[30] P. Pradhyumna, G. P. Shreya, and Mohana, "Graph neural network (GNN) in image and video understanding using deep learning for computer vision applications," in *Proceedings of the 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 1183–1189, Coimbatore, India, August 2021.

[31] R. K. Sinha, R. Pandey, and R. Pattnaik, "Deep learning for computer vision tasks: a review," in *Proceedings of the 2017 International Conference on Intelligent Computing and Control (I2C2)*, Coimbatore, India, June 2017.

[32] V. Rajesh, U. P. Naik, and Mohana, "Quantum Convolutional Neural Networks (QCNN) using deep learning for computer vision applications," in *Proceedings of the 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, pp. 728–734, Bangalore, India, August 2021.

[33] N. O'Mahony, S. Campbell, A. Carvalho et al., "Deep learning vs. traditional computer vision," in *Proceedings of the Science*

and Information Conference, pp. 128–144, Las Vegas, NV, USA, April 2019.

[34] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Exploiting multi-channels deep convolutional neural networks for multivariate time series classification," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 96–112, 2016.

[35] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," 2013, https://arxiv.org/abs/1301.3557.

[36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT press, Cambridge, 2016.

[37] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[38] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 160–167, Helsinki, Finland, July 2008.

[39] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with C-GAN synthetic images," *Computers and Electronics in Agriculture*, vol. 187, Article ID 106279, 2021.

[40] M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, "ToLeD: tomato leaf disease detection using convolution neural network," *Procedia Computer Science*, vol. 167, pp. 293–301, 2020.

[41] A. Elhassouny and F. Smarandache, "Smart mobile application to recognize tomato leaf diseases using convolutional neural networks," in *Proceedings of the 2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, pp. 1–4, Agadir, Morocco, July 2019.

[42] H. Durmus, E. O. Günes, and M. Kırcı, "Disease detection on the leaves of the tomato plants by using deep learning," in *Proceedings of the 2017 6th International Conference on AgroGeoinformatics*, pp. 1–5, Fairfax, VA, USA, August 2017.

[43] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.