IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 June 2022, accepted 12 July 2022, date of publication 19 July 2022, date of current version 27 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3192428

RESEARCH ARTICLE

Tomato Disease Recognition Using a Compact Convolutional Neural Network

EMRE ÖZBİLGE[1], MEHTAP KÖSE ULUKÖK[2], ÖNSEN TOYGAR[3], AND EBRU OZBİLGE[4]

[1]Computer Engineering Department, Faculty of Engineering, Cyprus International University, 99258 Nicosia, North Cyprus, Mersin 10, Turkey
[2]Computer Engineering Department, Faculty of Architecture and Engineering, Bahçeşehir Cyprus University, 99200 Nicosia, North Cyprus, Mersin 10, Turkey
[3]Computer Engineering Department, Faculty of Engineering, Eastern Mediterranean University, 99628 Famagusta, North Cyprus, Mersin 10, Turkey
[4]Department of Mathematics and Statistics, American University of the Middle East, Egaila 54200, Kuwait

Corresponding author: Emre Özbilge (eozbilge@ciu.edu.tr)

ABSTRACT The detection of diseases in tomatoes in advance and early intervention and treatment increase the production amount, efficiency, and quality, which will satisfy the consumer with a more affordable shelf price. Thus, the efforts of farmers waiting for harvests throughout the season are not wasted. In this study, a compact convolutional neural network (CNN) is proposed for a disease identification task in which the network comprises only six layers, which is why it is computationally inexpensive in terms of the parameters employed in the network. This network was trained using PlantVillage's tomato crop dataset, which consisted of 10 classes (nine diseases and one healthy). The proposed network was first compared with the well-known pre-trained ImageNet deep networks using a transfer learning approach. The results show that the proposed network performs better than pre-trained knowledge transferred deep network models, and that there is no need to constitute very large, complicated network architectures to achieve superior tomato disease identification performance. Furthermore, data augmentation techniques are employed during network training to improve the performance of the proposed network. The proposed network achieved an accuracy of the F_1 score, Matthews correlation coefficient, true positive rate, and true negative rate of 99.70%, 98.49%, 98.31%, 98.49%, and 99.81%, respectively, using 9,077 unseen test images. Our results are better than or similar to those of state-of-the-art deep neural network approaches that use the PlantVillage database and the proposed method employs the cheapest architecture.

INDEX TERMS Tomato disease classification, deep learning, computer vision, transfer learning, data augmentation.

I. INTRODUCTION

Plant disease identification is an important and useful task in agriculture since the early detection of diseases prevents the spread of the disease and finally the loss of the products [1]. Tomato, a nutrition-rich plant with a high source of income for farmers, is one of the most widely grown plants worldwide. However, various tomato plant diseases observed on leaves affect the products in terms of quantity and quality, and therefore, decrease productivity. Some well-known diseases of tomato leaves include mosaic virus, yellow leaf curl virus, target spot, two-spotted spider mite, septoria leaf spot, leaf mold, late blight, early blight, and bacterial spot, as shown in Figure 1.

The associate editor coordinating the review of this manuscript and approving it for publication was Easter Selvan Suviseshamuthu.

Recently, image processing and computer vision methods have been applied to tomato leaf images to detect tomato plant diseases. These approaches are used to extract leaf image features using handcrafted methods or deep learning architectures. Although hand-crafted techniques are successful in many identification problems, such as human identification, animal recognition, and plant disease classification [1]–[3], deep-learning-based approaches have been reported to achieve higher accuracy [4]–[7]. The new trend in tomato disease identification studies is to employ deep learning-based architectures which work well and exhibit high performance [8]–[10].

In this paper, a compact deep convolutional neural network (CNN) architecture is proposed to learn the relationship between a tomato leaf image and its corresponding disease, which is a classification task. One of the advantages of using

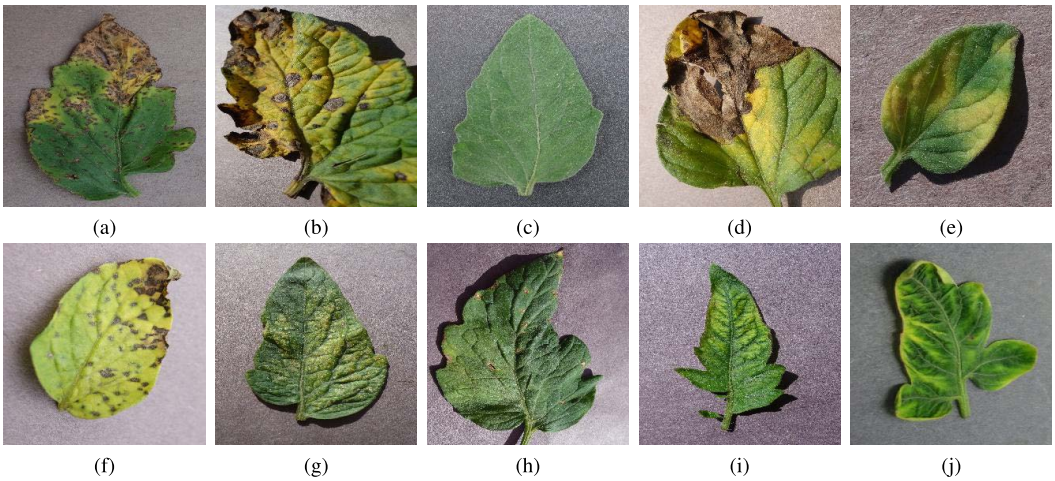

FIGURE 1. Sample images for each class of tomato disease. (a) Bacterial spot, (b) early blight, (c) healthy, (d) late blight, (e) leaf mold, (f) septoria leaf spot, (g) two-spotted spider mite, (h) target spot, (i) mosaic virus, (j) yellow leaf curl virus.

convolutional neural networks is that they automatically extract relevant features throughout their convolutional layers, making them a powerful classifier for computer vision-based tasks. The aim of this paper is to implement a robust deep network with high accuracy and at the same time with a computationally cheap architecture in terms of the number of parameters which are contained in the network in order to be used in the devices which have limited memory or processing capacity such as mobile phones and small quadcopter drones.

Experiments were conducted on tomato leaf images from the PlantVillage database using healthy and nine different disease types observed on tomato leaves. The results are encouraging, and a comparison with state-of-the-art methods shows that the proposed method performs better than other deep learning-based approaches. The main contributions of this study are as follows.

1) A compact deep CNN architecture with only six layers was proposed for tomato disease recognition.
2) It was also shown that there is no need for a very large deep network for the tomato disease classification problem, and which was analysed and proven by comparing the proposed deep network with the well-known ImageNet deep network models.
3) The proposed model with data augmentation using contrast and rotation achieves superior performance on all tomato disease classes and outperforms state-of-the-art deep learning-based approaches that employ the PlantVillage database for tomato disease classification.

The remainder of this paper is organised as follows. Section II explains previous research on the tomato disease classification problem. Section III presents the proposed deep convolutional neural network architecture for tomato leaf disease classification with a brief description of the training algorithm, training parameters, and architecture of the proposed network. Next, the PlantVillage dataset and the preparation of the dataset for network training and performance assessments of the network are described in Section IV. The experimental results are discussed in Section V with a statistical performance analysis of the proposed network and a comparison with state-of-the-art deep network architectures. Finally, Section VI presents the conclusions and highlights the future direction of this research.

II. RELATED WORK

Early detection of tomato plant diseases directly affects the quantity and quality of plant products. Several studies have been performed between 2010 and 2016 on tomato disease classification using different classifiers with texture, colour, or shape features of tomato leaves [1]. In general, SVM-based- or neural-network-based classifiers were employed in that period. Subsequently, deep learning-based computer vision approaches have been studied by many researchers, and they have shown that they provide a tool for automated tomato plant disease identification and diagnosis effectively, even if the experts may fail to identify them successfully.

[11] performed prediction of bacterial spot disease for tomato plants using visual spectrum images in laboratory conditions. They used a correlation coefficient spectrum, partial least squares (PLS) regression, and stepwise multiple linear regression (SMLR) procedure to identify important wavelengths for creating their predictive models. Images were taken approximately every 5 days for 22-47 samples. Although it can be considered an effective method to detect diseases, it is expensive and difficult to apply to large plant areas.

On the other hand, fast and accurate bacterial spot detection in tomato reduces the cost of treatment and monitoring. [12] developed an automatic detection of bacterial spot caused tomato disease in its early stage. Visual spectrum images were obtained using a commercial camera. The images were pre-processed and transformed into the CIELab colour space. Then, the clustering process is applied to channel a. The system was tested using 180 images.

CNN have become popular for plant disease identification over the last few years. This allows for the classification of large data with high accuracy. Many variants of CNN have been implemented by researchers for tomato plant disease diagnosis with the best detection accuracy of 99.53% [5]–[7], [13]. In this respect, [8] proposed a simplified and efficient CNN model with 8 hidden layers for tomato crop disease identification. The authors employed a lightweight CNN model to classify nine types of tomato crop diseases in the PlantVillage dataset. The highest accuracy achieved by the proposed method on the PlantVillage dataset is 98.4%.

Additionally, the Both-channel residual attention network (B-ARNet) model was recently proposed for tomato plant disease diagnosis [9]. Tomato leaf images were enhanced by a binary wavelet transform and then classified into 8,616 images with an overall detection accuracy of 89%.

Moreover, a recent study reported that a deep learning-based ResNet50 architecture on the PlantVillage tomato dataset detects previously unseen tomato plant diseases in its early stage with 97% maximum accuracy with colour images [10].

The state-of-the-art methods related to tomato disease identification indicates that deep learning based approaches achieve high performance compared to hand-crafted methods, however it still needs more research in order to improve the accuracy to the highest level and reduce computation time.

III. PROPOSED CNN MODEL

The proposed CNN model was developed with a more compact design than well-known deep networks which can be successfully implemented on the ImageNet database, such as Inception [14]–[16], ResNet [17], [18], VGG [19], MobileNet [20], [21], and Xception [22] architectures. The number of layers in these networks is large; therefore, such a network presents a very high degree of nonlinearity. These types of large networks are suitable for modelling the ImageNet database, which contains over 14 million data and 1000 classes. However, these networks may face overfitting, particularly when trying to model a smaller number of output classes, such as tomato leaf disease problems which have 10 distinct classes. The ImageNet database has various dissimilar input images, so such a large number of different images can only be classified with large deep networks that require a high nonlinearity to distinguish input images; as a result, the top performances on ImageNet are obtained with large deep networks [23]. On the other hand, the tomato leaf diseases database contains only leaves that consist of the variation of the green colour with infection regions (*i.e* the colour distribution of the leaf images are close to each other); thus, a much smaller network architecture can outperform the other architectures.

The proposed CNN model for tomato leaf disease classification is shown in Figure 2. The network receives $[256 \times 256 \times 3]$ pixel colour image as an input to the network. Each input image is first rescaled in the range of [0, 1] before being fed to the network. The rescaled image is then presented to the first convolutional layer of the network to perform a convolutional operation for each patch of the image. To determine the filter size on the convolutional layer, 32, 64, and 128 filters were tested, and the performance of 32 filters was found to be slightly better than that of the others. To overcome the covariate shift problem, after the convolutional operation, batch normalisation is applied before the rectified linear unit (ReLU) activation of each convolved node is computed. Subsequently, a maximum pooling operation was performed to downsample the acquired feature map and condense the most relevant features in the patches. These operations are repeated using other convolutional and maximum-pooling layers. The outputs from the final pooling layer are first converted to a single column vector (*i.e* flattened), and then this vector is connected to a fully connected dense layer. The fully connected layer consists of 100 hidden nodes, and in this layer, the sum of the weighted input data from the flattened layer is first batch-normalised. Subsequently, the ReLU activation of each node was computed. Finally, this layer was connected to a softmax layer with 10 output nodes (*i.e* the number of leaf diseases), and each node represented the probability of the classes for the current input image. To determine the predicted class of the input image, the maximum probability among the output nodes is selected, as given in (1):

$$\hat{y}(\mathbf{X}) = \arg\max_{i \in [1,c]} (a_i), \qquad a_i = \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}}, \tag{1}$$

where $\hat{y}$ is the index of the output node with the maximum probability, $\mathbf{X}$ indicates the current input image, a_i is the activation value of the output node i, z_i is the sum of the weighted inputs to the output node i and c is the number of output nodes.

The proposed network comprises 831,110 parameters, and only 830,782 parameters can be trained by the backpropagation algorithm. Such a large number of parameters of a network are prone to overfitting during the training; as a result, the network's performance on unseen tomato leaf images can be poor. To overcome this, dropout regularisation is used from the flattened data vector to the fully connected layer and from the fully connected layer to the softmax layer, as shown in Figure 2. This approach randomly disables some nodes on the specified layers, depending on the predefined dropout rate along each input image presented to the network. As a result, the weight balance of the network is spread to all weight connections so that the network does not rely on one feature during training.

It is also important to select an appropriate learning rate in order to converge the local minimum to optimise the cost function (*i.e* too large a learning rate may diverge the learning or too small a rate may slow the convergence). Consequently, the initial learning rate α_0 is found to be a good choice to set 0.01 and reduces the learning rate exponentially in every epoch; therefore, when the learning is close to the local

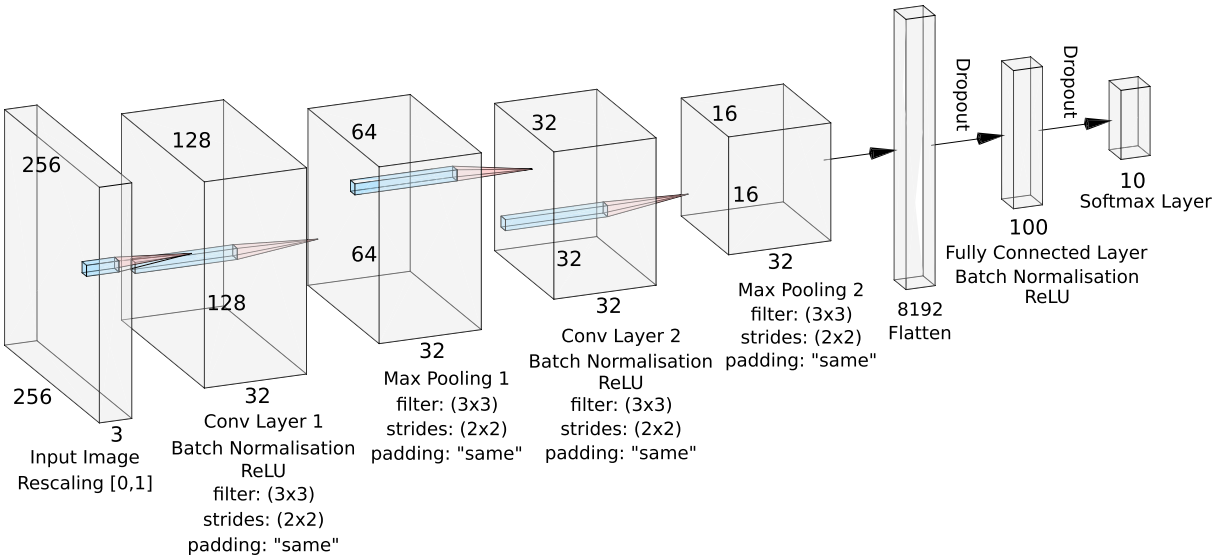

FIGURE 2. Proposed compact CNN model.

minimum, the algorithm performs smaller steps to reach the minimum. The learning rate is expressed as follows:

$$\alpha = \alpha_0 e^{-\tau t} \tag{2}$$

where τ is the decay rate, and t indicates the current iteration.

Furthermore, the momentum and RMSprop parameters were set alongside the learning rate to remove the oscillation on the gradient descent and speed up the training algorithm to reach local minima.

Finally, the backpropagation algorithm is used to propagate the loss, calculated at the softmax layer as given in (3) throughout the first convolutional layer by computing the gradient of the loss function with respect to the network's weights. The loss is given as:

$$\xi(\mathbf{T}, \mathbf{A}) = \sum_{i=1}^{n} \xi(\mathbf{t}_i, \mathbf{a}_i) = -\sum_{i=1}^{n} \sum_{j=1}^{c} t_{ij} \cdot \log(a_{ij}) \tag{3}$$

where $\mathbf{T} = [\mathbf{t}_1 \mathbf{t}_2 \cdots \mathbf{t}_n]$ indicates the target output vectors of the input images, where each target vector $\mathbf{t}_i = [t_1 t_2 \cdots t_c]$ is defined as a one-hot vector (*i.e* all elements of the vector are set to zero except the target output node corresponding to the class of the current input image, which is one), and n is the number of data in the current batch.

A summary of the training parameters for the proposed CNN model used in the experiments is presented in Table 1.

IV. EXPERIMENTAL METHODS

This section describes PlantVillage's tomato dataset and data preparation for network training, and provides a detailed explanation of the performance evaluation metrics of the network used in the experiments.

TABLE 1. Training parameters for the proposed CNN model.

Parameter	Description
Optimisation algorithm	Adam optimiser
Initial learning rate (α_0)	0.01
Decay rate (τ)	0.05
Weight initialisation	Xavier initialiser
Momentum	0.9
RMSprop	0.999
Batch size	32
Number of epoch	100
Dropout rate	0.5

TABLE 2. The total numbers of training, validation, and test images were used for each class.

Class ID	Class Label	Total Data	Training and Validation Data	Test Data
C0	Bacterial Spot	2127	1064	1063
C1	Early Blight	1000	500	500
C2	Healthy	1591	796	795
C3	Late Blight	1909	955	954
C4	Leaf Mold	952	476	476
C5	Septoria Leaf Spot	1771	886	885
C6	Two Spotted Spider Mite	1676	838	838
C7	Target Spot	1404	702	702
C8	Mosaic Virus	373	187	186
C9	Yellow Leaf Curl Virus	5357	2679	2678
		Total:	9083 (8175:908)	9077

A. DATASET AND DATA PREPARATION

The proposed CNN model was trained and tested using the PlantVillage dataset [24] which contains 54,305 leaf images

TABLE 3. Measures of the network classification performance for each class and overall performance.

Measure for class i	Formula	Weighted-Average
ACC_i	$\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}$	$\frac{\sum_{i=0}^{c} n_i \cdot ACC_i}{n}$
F_{1i}	$\frac{2 \cdot TP_i}{2 \cdot TP_i + FP_i + FN_i}$	$\frac{\sum_{i=0}^{c} n_i \cdot F_{1i}}{n}$
MCC_i	$\frac{TP_i \cdot TN_i - FP_i \cdot FN_i}{\sqrt{(TP_i + FP_i) \cdot (TP_i + FN_i) \cdot (TN_i + FP_i) \cdot (TN_i + FN_i)}}$	$\frac{\sum_{i=0}^{c} n_i \cdot MCC_i}{n}$
TPR_i	$\frac{TP_i}{TP_i + FN_i}$	$\frac{\sum_{i=0}^{c} n_i \cdot TPR_i}{n}$
TNR_i	$\frac{TN_i}{TN_i + FP_i}$	$\frac{\sum_{i=0}^{c} n_i \cdot TNR_i}{n}$

from 14 crops with various healthy and diseased leaf datasets. In this study, only tomato crops were considered which consisted of nine distinct leaf disease classes and one healthy class, with a total of 18,160 images.

Each class of the tomato dataset was divided equally into two halves. The first half of each class was then combined for the training and validation of the deep network models during training. The data for the training and validation were divided in a 90:10 ratio; thus, 8,175 image data were used for training and the remaining 908 image data were used to validate the model during training. Thus, after each epoch of training, the current acquired network model was tested on the validation set for model evaluation. During the experiments, different training-validation split rates (*i.e* 90:10, 80:20, and 70:30 ratios) were also tested, and it was found that increasing the number of validation data did not improve the generalisation performance of the model on the unseen test images. The second half of the database which is also called the test dataset, was used to verify the overall performance of the acquired models after the training process of the models was completed. Statistical assessments on the test dataset were considered for a comparison study of the deep network models described in this paper. Table 2 summarises the amount of data belonging to each class of tomato leaf diseases and indicates the size of the training, validation, and test data used from each class. There is an imbalanced set of data contained between the classes, which might be a problem for neural network training because of the lack of data on a class that can cause the corresponding class to be not learned accurately by the network [25].

B. PERFORMANCE EVALUATION METRICS

To evaluate the performance of the proposed network and compare it with other deep networks, a $c \times c$ multiclass confusion matrix was constructed. True positive (TP), true negative (TN), false positive (FP), and false negative (FN) values were then computed for each class on the matrix. Once these values were computed, the accuracy (*ACC*), F_1 score, Matthews correlation coefficient (*MCC*), true positive rate (*TPR*), and true negative rate (*TNR*) were computed for each class, as given in Table 3. Here, the most informative and accurate statistical metric is *MCC* according to [26]; therefore, *MCC* outcomes are considered as the main criterion for the evaluation of the network model. Consequently, a value close to one for *MCC* indicates a strong correlation between the actual and predicted classes of the network. One issue with the available PlantVillage dataset is that the amount of data in each class is unbalanced. To overcome this, the weighted average value of each statistical metric using all classes was computed to determine the overall performance of the network.

V. EXPERIMENTAL RESULTS

This section describes the performance of the proposed CNN model on PlantVillage's tomato crops, compares the proposed network with well-known deep networks, and demonstrates a method to improve the generalisation of the proposed model.

A. PERFORMANCE OF THE PROPOSED CNN MODEL

After training the proposed CNN model, the obtained model was verified on the unseen test dataset, and Table 4 illustrates the percentage of correctly classified tomato leaf images by the proposed network. Almost all tomato leaf diseases were classified correctly over 95%, except for class C1 (early blight disease). As can be seen clearly in the table, the network misclassifies class C1 with classes C0 (bacterial spot disease) and C2 (healthy) at 2.6% and 5.2%, respectively. A similar performance was also obtained when testing the model using the validation dataset by using the trained network model. Consequently, the network has a slightly high-variance problem (*i.e* overfitting) for predicting class C1. This also shows that the training and test data originate different distributions. To overcome this issue, more data are required [27] to improve the classification performance of the network for predicting class C1, and as addressed in Section V-C. In fact, mosaic virus disease has less training data available, as seen in Table 2; however, the image features

TABLE 4. Normalised confusion matrix for each class using the proposed model on the PlantVillage dataset.

		Predicted									
		C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
Actual	C0	**98.12**	0.09	0.19	0	0	0	0.38	1.22	0	0
	C1	2.6	**87.4**	5.2	0.4	1.2	0.6	1.8	0.2	0.6	0
	C2	0	1.57	**96.44**	0.84	0.21	0.1	0.21	0.31	0	0.31
	C3	0.21	0	1.05	**97.06**	0.84	0.63	0.21	0	0	0
	C4	0.34	0.23	0.68	0.79	**97.85**	0	0.11	0	0	0
	C5	0	0	0	0	0	**98.69**	0.84	0.36	0	0.12
	C6	0.14	0.28	0	0	1	3.13	**94.87**	0.14	0	0.43
	C7	0.22	0	0	0.04	0	0.04	0	**99.7**	0	0
	C8	0	0	0	1.08	0.54	1.08	0	0	**97.31**	0
	C9	0	0	0	0	0	0	0	0	0	**100**

of mosaic virus might be more distinguishable by the network during the training.

Deep networks do not produce a transparent model [28] for understanding how the extracted features contribute to the final prediction. This is because deep networks contain many hidden layers, which present high non-linearity, making it impossible to understand the models very clearly in the mathematical perceptive. To slightly understand the behaviour of the classifier, gradient-weighted class activation mapping (Grad-CAM) analysis was performed. This analysis highlights the region of interest (ROI) in the input image when processing by the specified convolutional layer; thus, it might provide a level of understandable information about the extracted features that activate the classifier to predict the associated output class. Here, the important features from the final convolutional layer are used in the Grad-CAM analysis because the higher convolutional layers detect more specialised features instead of detecting edges in the lower convolutional layers. Figure 3 shows the Grad-CAM visualisation for randomly selected test images from each class of tomato disease. As can be seen clearly in Figures 3a, 3b, 3d to 3f and 3h to 3j, the most activated areas contain abnormalities on the leaves that also indicate non-uniform patterns on the images that are more-likely to be extracted during the convolutional operations. Each leaf carries a distinct abnormality; therefore, the extracted information becomes more distinguishable by the classifier. On the other hand, there was no abnormal area on the healthy leaf; consequently, the highest values were obtained outside the leaf, as shown in Figure 3c. This is also a distinguishable property of healthy leaves in comparison with other activated areas that belong to diseased leaves. Interestingly, the network yielded lower activation values for the entire input image of the two-spotted spider mite disease and did not find the area of the disease interesting enough, as shown in Figure 3g. In fact, the lowest activations can be separated by the classifier so that almost 95% of the two-spotted spider mite diseased images are classified correctly (*i.e* class C6 in Table 4).

B. COMPARISON OF PROPOSED CNN MODEL WITH PRE-TRAINED ImageNet MODELS

Further analysis was also carried out to evaluate the proposed CNN model with well-known pretrained deep network architectures. A transfer learning strategy was applied to train such large deep networks. These networks have been trained using data from the ImageNet challenge and have already learned the features of leaf images (this is because of the existence of a huge image database). Therefore, the knowledge acquired from this database is transferred to the tomato leaf disease task. To do this, the layers before the fully connected layer at the pre-trained network weights are frozen which is called the base model, and the base model is connected to a new fully connected layer and softmax layer to learn the new task (*i.e* tomato leaf diseases). Figure 4 shows the architecture of transfer learning for the pre-trained models, where the outputs (extracted features) of the base (pre-trained) model are connected to a new fully connected layer that contains 40 hidden nodes. This layer was followed by a batch normalisation operation, and the ReLU activation outputs were computed. Finally, the outputs from the hidden layer were connected to 10 softmax output nodes. The training parameters of these pre-trained networks were the same as those defined in Table 1.

The performance of each pretrained deep network is presented in Table 5. The results show that ResNet152 performed the worst on the unseen test dataset. This network consists of with 152 deep layers and it generalises the unseen image data poorly; therefore, it is found that the network has a high bias issue (*i.e* underfitting), as reported with a low *MCC* value, *i.e* $MCC = 0.6122$. In contrast, the best performance using a pre-trained network was obtained with the DenseNet201 architecture, where *MCC* value was 0.9713.

The main benefit of using a very deep network is that it is possible to represent a very complex nonlinear function by extracting complex features from deeper layers. However, this type of deeper network can be struggling with the vanishing gradient problem when the gradient of the final

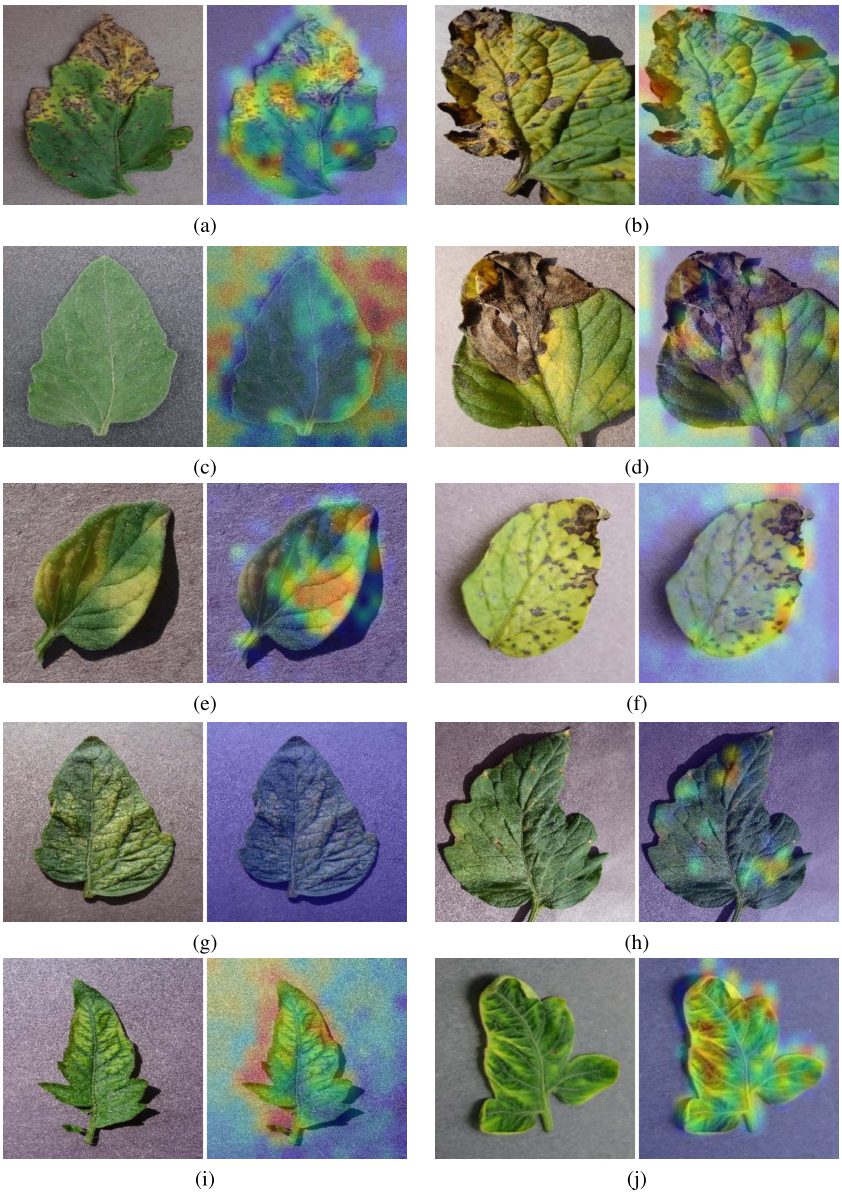

FIGURE 3. Visualisation of Grad-CAM analysis. The left image shows the original image, and the right image shows the ROI in the original image highlighted by the final convolutional layer for the (a) bacterial spot, (b) early blight, (c) healthy, (d) late blight, (e) leaf mold, (f) septoria leaf spot, (g) two-spotted spider mite, (h) target spot, (i) mosaic virus, (j) yellow leaf curl virus.

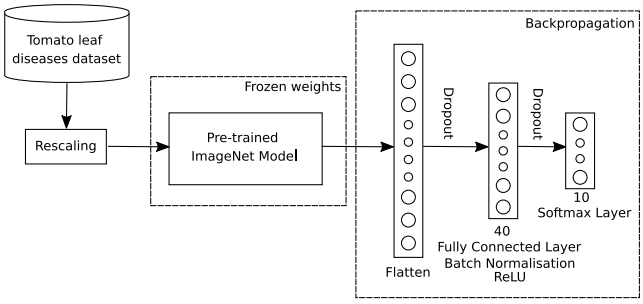

FIGURE 4. Transfer learning with pre-trained ImageNet models.

layer is backpropagated until the first layer. To overcome this problem, a residual connection (or skip connection) was introduced with the ResNet architecture [17]. Here, the residual block in the network is used to propagate a copy of the residual block's input that forwards it to the end of the residual block without processing it. In contrast to the DenseNet architecture, each layer receives input features from all preceding layers, and it is also claimed that DenseNet is less prone to the vanishing gradient problem because it backpropagates the error signals from the final layer to the first layer in a more direct manner [29]. Therefore, DenseNet yields richer and diversified features which make a stronger contribution to distinguishing the desired output classes of tomato diseases [30]. The difference between the two architectures are illustrated in Figure 5.

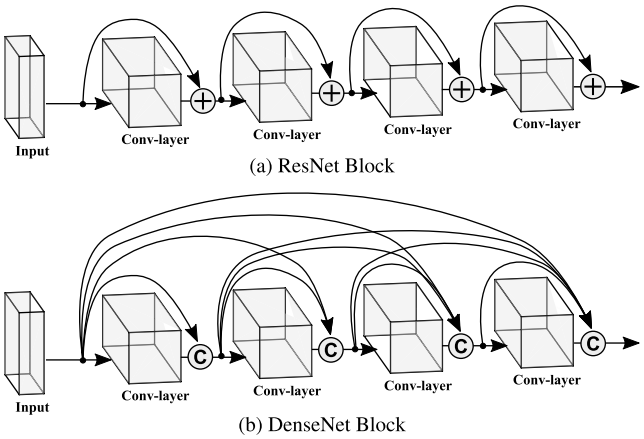

FIGURE 5. Simplified versions of ResNet and DenseNet blocks. ResNet uses element-wise addition, whereas DenseNet uses channel-wise concatenation to combine the features. All intermediate layers were removed for clarity. The detailed architectures can be found in [17], [29].

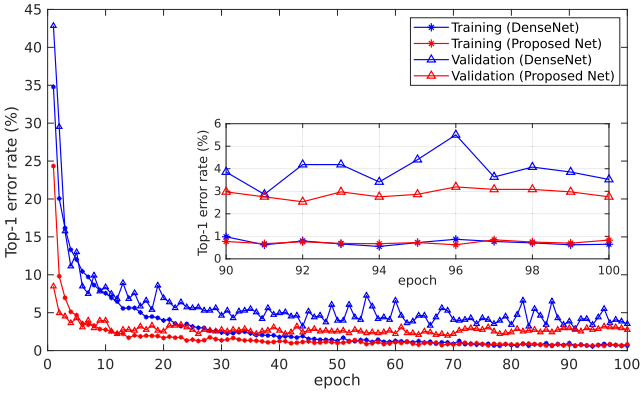

FIGURE 6. Top-1 error rates of both DenseNet201 and the proposed models for the training and validation datasets during the training epochs. The inner graph clearly shows the last ten epochs.

The proposed compact CNN model performed slightly better than the DenseNet201 model in all statistical evaluations, as shown in Table 5. The results show that the proposed model predicts more accurate tomato diseases than the DenseNet201 model, where *TPR* value of the proposed model is higher; thus, approximately 0.29% more diseases are classified correctly by the smaller inexpensive deep network. Furthermore, the learning curves of both models were also compared, where the DenseNet201 model reached the same training error as the proposed model at around epoch 45 during the training process, as shown in Figure 6. The top-1 error rates of the DenseNet201 model at the end of training which were obtained for the training and validation datasets, were reported as 0.6483% and 3.5240%, respectively. Therefore, a variance of 2.8757% between the training and validation images is obtained for the DenseNet201 model. However, the variance of the proposed model was lower than that of the DenseNet201 model, where the variance was computed as 1.9212%. Both models have small variances, so this also shows low bias and low variance cases which yield good models for identifying tomato diseases.

TABLE 5. Performance of pretrained ImageNet models and comparison with the proposed CNN model (based on 9,077 unseen test images).

Network Architecture	ACC	F_1	MCC	TPR	TNR
Xception	0.9850	0.9230	0.9146	0.9239	0.9900
VGG16	0.9881	0.9341	0.9275	0.9346	0.9931
VGG19	0.9851	0.9228	0.9145	0.9239	0.9903
ResNet50	0.9366	0.7060	0.6779	0.7308	0.9513
ResNet101	0.9343	0.7016	0.6729	0.7244	0.9546
ResNet152	0.9141	0.6522	0.6122	0.6825	0.9269
ResNet50V2	0.9871	0.9326	0.9256	0.9332	0.9914
ResNet101V2	0.9861	0.9323	0.9245	0.9332	0.9895
ResNet152V2	0.9894	0.9422	0.9367	0.9429	0.9933
InceptionV1	0.9825	0.9075	0.8988	0.9088	0.9894
InceptionV3	0.9827	0.9104	0.9008	0.9115	0.9883
InceptionResNetV2	0.9843	0.9188	0.9098	0.9197	0.9894
MobileNet	0.9919	0.9549	0.9504	0.9554	0.9946
MobileNetV2	0.9889	0.9403	0.9341	0.9408	0.9930
MobileNetV3Small	0.9923	0.9573	0.9531	0.9578	0.9950
MobileNetV3Large	0.9926	0.9606	0.9565	0.9609	0.9951
DenseNet121	0.9949	0.9714	0.9686	0.9716	0.9969
DenseNet169	0.9947	0.9716	0.9688	0.9719	0.9964
DenseNet201	**0.9954**	**0.9737**	**0.9713**	**0.9740**	**0.9970**
NASNetMobile	0.9838	0.9128	0.9043	0.9145	0.9896
NASNetLarge	0.9850	0.9199	0.9120	0.9204	0.9906
EfficientNetB0	0.9937	0.9667	0.9632	0.9669	0.9960
EfficientNetB1	0.9930	0.9617	0.9579	0.9621	0.9954
EfficientNetB2	0.9932	0.9636	0.9598	0.9639	0.9958
EfficientNetB3	0.9932	0.9641	0.9603	0.9644	0.9955
EfficientNetB4	0.9926	0.9605	0.9563	0.9607	0.9953
EfficientNetB5	0.9917	0.9572	0.9525	0.9575	0.9944
EfficientNetB6	0.9919	0.9565	0.9521	0.9569	0.9950
EfficientNetB7	0.9924	0.9598	0.9557	0.9602	0.9951
EfficientNetV2B0	0.9938	0.9663	0.9628	0.9665	0.9959
EfficientNetV2B1	0.9941	0.9682	0.9649	0.9684	0.9962
EfficientNetV2B2	0.9935	0.9650	0.9613	0.9652	0.9956
EfficientNetV2B3	0.9945	0.9707	0.9677	0.9709	0.9965
EfficientNetV2S	0.9936	0.9670	0.9634	0.9673	0.9955
EfficientNetV2M	0.9911	0.9535	0.9485	0.9539	0.9940
EfficientNetV2L	0.9928	0.9616	0.9576	0.9619	0.9953
Proposed CNN	**0.9955**	**0.9767**	**0.9741**	**0.9769**	**0.9970**

The performances of both models for predicting each class of tomato diseases are presented in Table 6. As can be clearly seen, *MCC* values of each class for the proposed model are all greater than 0.9, and only the predictions of classes C0, C2, and C7 are slightly lower than that of the DenseNet201 model. In addition, the lowest value is obtained when the proposed model predicts class C1 (*i.e* early blight disease), as mentioned previously, because of the lack of a training dataset for class C1. Although both models exhibit very similar performances, the proposed model is computationally less expensive, where the DenseNet201 model contains

TABLE 6. Performance of DenseNet201 and the proposed CNN model for each class.

	DenseNet201					Proposed CNN				
	ACC	F_1	MCC	TPR	TNR	ACC	F_1	MCC	TPR	TNR
C0	0.9956	0.9814	0.9789	0.9906	0.9963	0.9952	0.9793	0.9766	0.9812	0.9970
C1	0.9899	0.9025	0.8990	0.8520	0.9979	0.9909	0.9133	0.9095	0.8740	0.9977
C2	0.9922	0.9634	0.9592	0.9790	0.9937	0.9920	0.9618	0.9573	0.9644	0.9952
C3	0.9955	0.9565	0.9542	0.9475	0.9981	0.9963	0.9645	0.9626	0.9706	0.9977
C4	0.9939	0.9690	0.9657	0.9718	0.9963	0.9957	0.9780	0.9756	0.9785	0.9976
C5	0.9944	0.9702	0.9673	0.9893	0.9949	0.9953	0.9747	0.9721	0.9869	0.9961
C6	0.9916	0.9448	0.9405	0.9259	0.9971	0.9934	0.9569	0.9534	0.9487	0.9971
C7	0.9983	0.9972	0.9960	0.9981	0.9984	0.9968	0.9946	0.9923	0.9970	0.9967
C8	0.9983	0.9589	0.9582	0.9409	0.9996	0.9991	0.9784	0.9779	0.9731	0.9997
C9	0.9982	0.9900	0.9890	0.9937	0.9987	0.9992	0.9956	0.9952	1.0000	0.9992

TABLE 7. Performance of the proposed CNN model with data augmentation techniques.

Proposed CNN	ACC	F_1	MCC	TPR	TNR
No Augmentation	0.9955	0.9767	0.9741	0.9769	0.9970
Contrast	0.9956	0.9767	0.9742	0.9769	0.9971
Rotation	0.9939	0.9668	0.9638	0.9671	0.9965
Contrast & Rotation	**0.9970**	**0.9849**	**0.9831**	**0.9849**	**0.9981**

23,237,794 parameters, whereas the 831,110 parameters are involved in the proposed model.

C. DATA AUGMENTATION TO IMPROVE GENERALISATION OF THE PROPOSED NETWORK

To improve the performance of the proposed model for poor predictions of class C1 (*i.e* early blight disease), the amount of training data was increased by generating a slightly modified version of the available training data. The extracted features from the final convolutional layer of the proposed network, for only the image data that cause the network to fail to predict class C1, demonstrate strong similarity with class C2 (*i.e* healthy leaves). This occurs whenever the number of disease indications is very low on the leaves; therefore, this information can be lost while passing through the convolution process.

Data augmentation is performed during training and is expected to reduce the overfitting issue [31], [32]. Two techniques were used to augment the image in the proposed model training: *rotation* and *contrast*. During the training, before each image is presented to the network, the original image is first rotated clockwise or counterclockwise randomly in the range of $\pm 36°$ (*i.e* 10% of one full rotation). The main reason for keeping the rotation amount small is that there is no extreme rotation difference between the training and test images. Otherwise, instead of reducing overfitting, it is possible to train the model with completely different images; therefore, the model does not predict well on the unseen test images. After obtaining the rotated image, its contrast also changes randomly to $\pm 10\%$. The contrast amount is also kept small because increasing the contrast of the image can also cause an increase in the dissimilarity between the training and test images.

The performance of the proposed CNN models using data augmentation techniques during training is given in Table 7. It is important to note that all the training parameters were kept the same for all the models, as defined in Table 1. Interestingly, applying contrast and rotation separately to augment a new image for model training did not significantly improve the generalisation performance of the model. *MCC* values using contrast and rotation individually are computed as 0.9742 and 0.9638, respectively; however, without using both data augmentation techniques, this value is 0.9741. However, when both augmentation techniques are used together, the results show that there is a slight improvement in the model. The network produces more confident predictions; in other words, it predicts a higher probability for diseases which are indicated with the smallest loss ξ of test images. In addition, the data augmented model predicts 98.49% (*TPR*) of diseases correctly; in contrast, this assessment is 97.69% for the model without data augmentation.

In fact, the proposed CNN model has already performed well (more than 94%) except for class C1 (early blight disease) without using augmented images in the training (as shown in Table 4); only class C1 (early blight disease) is predicted to be 87.4%. Thus, the overall *TPR* increases by only 0.8%. However, there was a significant improvement in predicting class C1, as shown in Table 8 where *TPR* value increased from 0.8740 to 0.9440. In addition, the network's performance in predicting other classes is also slightly improved; in contrast, classes C3 and C5 drop slightly. Finally, the normalised confusion matrix also shows the percentage of correctly classified images in each class, and almost all classes were classified correctly over 95%, as given in Table 9. Therefore, it can be stated that the proposed CNN model with data augmentation (contrast and

TABLE 8. Performance of proposed CNN model for each class with and without data augmentation techniques.

	Proposed CNN					Proposed CNN (contrast&rotation)				
	ACC	F_1	MCC	TPR	TNR	ACC	F_1	MCC	TPR	TNR
C0	0.9952	0.9793	0.9766	0.9812	0.9970	0.9972	0.9882	0.9866	0.9821	0.9993
C1	0.9909	0.9133	0.9095	0.8740	0.9977	0.9954	0.9574	0.9551	0.9440	0.9984
C2	0.9920	0.9618	0.9573	0.9644	0.9952	0.9941	0.9717	0.9683	0.9706	0.9968
C3	0.9963	0.9645	0.9626	0.9706	0.9977	0.9977	0.9777	0.9765	0.9664	0.9994
C4	0.9957	0.9780	0.9756	0.9785	0.9976	0.9978	0.9888	0.9875	0.9932	0.9983
C5	0.9953	0.9747	0.9721	0.9869	0.9961	0.9969	0.9833	0.9816	0.9857	0.9981
C6	0.9934	0.9569	0.9534	0.9487	0.9971	0.9947	0.9660	0.9632	0.9715	0.9967
C7	0.9968	0.9946	0.9923	0.9970	0.9967	0.9979	0.9965	0.9950	0.9978	0.9980
C8	0.9991	0.9784	0.9779	0.9731	0.9997	0.9996	0.9892	0.9890	0.9892	0.9998
C9	0.9992	0.9956	0.9952	1.0000	0.9992	0.9986	0.9919	0.9911	1.0000	0.9984

TABLE 9. Normalised confusion matrix of the proposed CNN model using data augmentation techniques (contrast and rotation).

		Predicted									
		C0	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**
Actual	**C0**	**98.21**	0.09	0.28	0	0	0	0.66	0.75	0	0
	C1	0.4	**94.4**	3.2	0	0.4	0	0.8	0.4	0.4	0
	C2	0	1.05	**97.06**	0.21	0.42	0.1	0.63	0.1	0	0.42
	C3	0	0	1.05	**96.64**	1.05	0.42	0.63	0.21	0	0
	C4	0.11	0.11	0.23	0.23	**99.32**	0	0	0	0	0
	C5	0	0.12	0	0	0	**98.57**	0.95	0.12	0	0.24
	C6	0	0.14	0	0	0.28	1.42	**97.15**	0	0	1
	C7	0.11	0	0	0	0	0.11	0	**99.78**	0	0
	C8	0	0	0	0.54	0.54	0	0	0	**98.92**	0
	C9	0	0	0	0	0	0	0	0	0	**100**

TABLE 10. Comparison with state-of-the-art tomato disease identification using the PlantVillage dataset.

Study	Year	Images	No. of Diseases	Methods	Accuracy	Parameters (Millions)
Rangarajan et al. [36]	2018	13,262	7	AlexNet	97.49%	62.3
Agarwal et al. [8]	2020	18,160	10	Custom CNN	98.70%	0.208
Suryawati et al. [33]	2018	18,160	10	VGG16	95.24%	138.4
Ashqar and Abu-Naser [35]	2018	9,000	6	Custom CNN	99.84%	3.6
Brahimi et al. [37]	2017	14,828	9	InceptionV1	99.18%	5.3
Elhassouny and Smarandache [34]	2019	7,176	10	MobileNet	90.30%	4.3
Wspanialy and Moussa [10]	2020	18,005	10	ResNet50	97%	25.6
Gonzalez-Huitron et al. [38]	2021	18,160	10	Xception	99.95%	22.9
Thangaraj et al. [39]	2021	16,578	10	Xception	99.55%	22.9
Proposed Model	**2022**	**18,160**	**10**	**Custom CNN**	**99.70%**	0.83

rotation) achieved superior performance in all tomato disease classes.

The proposed model was also compared with state-of-the-art deep neural network models for tomato disease classification, as shown in Table 10. It is clear that apart from this study, there are four other studies [8], [10], [33], [34] modelling the same number of disease classes, and the results show that the proposed model performed better than the other models. Although the accuracy of the model described in [35] is the highest (only 0.14% better than the proposed

model), six tomato diseases were modelled which does not indicate whether this model can perform similarly when there are 10 disease classes. Furthermore, the number of network weights in the proposed model was less than that in the other models.

VI. CONCLUSION

In this study, a compact deep neural network architecture is proposed for the classification of tomato leaf diseases. This network consists of only six layers in comparison with well-known pre-trained knowledge-transferred ImageNet deep network models. The proposed network was trained to learn about nine distinct diseases and healthy leaves using the PlantVillage database. To evaluate the performance of the network, various statistical assessments were performed using 9,077 unseen test images. The proposed model is first compared with various pre-trained deep networks, and the results show that the proposed model performs better than the pre-trained models because the pre-trained models have a large number of layers and are more prone to overfitting. The DenseNet201 pre-trained model performs very closely to the proposed compact model; however, the proposed model contains significantly fewer parameters in its architecture. For the tomato disease identification problem, it is important to have a small model that can be used on small devices such as drones for automatic disease identification by flying on a very large crop field. Additionally, data augmentation methods are employed first to identify which technique actually improves the network performance, and second, to increase the number of images for classes which have inadequate data during training. In fact, adding some noise to the raw images using data augmentation techniques actually improves the generalisation performance of the obtained model when used in different crop environments, where the captured images might be under different lighting conditions, weather conditions, or images at different orientations and distances in comparison with the trained data. Consequently, the proposed model was improved and made more reliable and strong predictions. Moreover, the overall performance of the proposed model is also improved in this manner; therefore, the overall accuracy and true positive prediction rate of the proposed model increased from 99.55% to 99.70% and from 97.69% to 98.49%, respectively. In conclusion, it can be stated that there is no need to use very deep architectures which are computationally expensive for the tomato disease classification problem.

REFERENCES

[1] S. Kaur, S. Pandey, and S. Goel, "Plants disease identification and classification through leaf images: A survey," *Arch. Comput. Methods Eng.*, vol. 26, no. 2, pp. 507–530, 2019.

[2] M. Farmanbar and Ö. Toygar, "A hybrid approach for person identification using palmprint and face biometrics," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 29, no. 6, Sep. 2015, Art. no. 1556009.

[3] S. Taheri and Ö. Toygar, "Animal classification using facial images with score-level fusion," *IET Comput. Vis.*, vol. 12, no. 5, pp. 679–685, Aug. 2018.

[4] A. K.-F. Lui, Y.-H. Chan, and M.-F. Leung, "Modelling of destinations for data-driven pedestrian trajectory prediction in public buildings," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 1709–1717.

[5] S. H. Lee, H. Goëau, P. Bonnet, and A. Joly, "New perspectives on plant disease characterization based on deep learning," *Comput. Electron. Agricult.*, vol. 170, Mar. 2020, Art. no. 105220.

[6] H. Nazki, S. Yoon, A. Fuentes, and D. S. Park, "Unsupervised image translation using adversarial networks for improved plant disease recognition," *Comput. Electron. Agricult.*, vol. 168, Jan. 2020, Art. no. 105117.

[7] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agricult.*, vol. 145, pp. 311–318, Feb. 2018.

[8] M. Agarwal, S. K. Gupta, and K. K. Biswas, "Development of efficient CNN model for tomato crop disease identification," *Sustain. Comput., Informat. Syst.*, vol. 28, Dec. 2020, Art. no. 100407.

[9] X. Chen, G. Zhou, A. Chen, J. Yi, W. Zhang, and Y. Hu, "Identification of tomato leaf diseases based on combination of ABCK-BWTR and B-ARNet," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105730.

[10] P. Wspanialy and M. Moussa, "A detection and severity estimation system for generic diseases of tomato greenhouse plants," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105701.

[11] C. D. Jones, J. B. Jones, and W. S. Lee, "Diagnosis of bacterial spot of tomato using spectral signatures," *Comput. Electron. Agricult.*, vol. 74, no. 2, pp. 329–335, Nov. 2010.

[12] D. L. Borges, S. T. C. de M. Guedes, A. R. Nascimento, and P. Melo-Pinto, "Detecting and grading severity of bacterial spot caused by Xanthomonas spp. in tomato (solanum lycopersicon) fields using visible spectrum images," *Comput. Electron. Agricult.*, vol. 125, pp. 149–159, Jul. 2016.

[13] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Comput. Electron. Agricult.*, vol. 153, pp. 46–53, Oct. 2018.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2818–2826.

[16] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 630–645.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[22] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[23] (2021). PapersWithCode. *Image Classification on ImageNet*. Accessed: Jan. 2, 2021. [Online]. Available: https://paperswithcode.com/sota/image-classification-on-imagenet

[24] D. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," 2015, *arXiv:1511.08060*.

[25] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[26] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, 2020.

[27] A. Ng. (2019). *Machine Learning Yearning: Technical Strategy for AI Engineers in the Era of Deep Learning*. [Online]. Available: https://www.mlyearning.org
[28] S. Billings and H.-L. Wei, "NARMAX model as a sparse, interpretable and transparent machine learning approach for big medical and healthcare data analysis," in *Proc. IEEE 21st Int. Conf. High Perform. Comput. Commun.; IEEE 17th Int. Conf. Smart City; IEEE 5th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Aug. 2019, pp. 2743–2750.
[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[30] C. Zhou, S. Zhou, J. Xing, and J. Song, "Tomato leaf disease identification by restructured deep residual dense network," *IEEE Access*, vol. 9, pp. 28822–28831, 2021.
[31] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.
[32] Q. Wu, Y. Chen, and J. Meng, "DCGAN-based data augmentation for tomato leaf disease identification," *IEEE Access*, vol. 8, pp. 9716–9828, 2020.
[33] E. Suryawati, R. Sustika, R. S. Yuwana, A. Subekti, and H. F. Pardede, "Deep structured convolutional neural network for tomato diseases detection," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS)*, Oct. 2018, pp. 385–390.
[34] A. Elhassouny and F. Smarandache, "Smart mobile application to recognize tomato leaf diseases using convolutional neural networks," in *Proc. Int. Conf. Comput. Sci. Renew. Energies (ICCSRE)*, Jul. 2019, pp. 1–4.
[35] B. A. Ashqar and S. S. Abu-Naser, "Image-based tomato leaves diseases detection using deep learning," *Int. J. Academic Eng. Res.*, vol. 2, no. 12, pp. 10–16, 2018.
[36] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Proc. Comput. Sci.*, vol. 133, pp. 1040–1047, Jan. 2018.
[37] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: Classification and symptoms visualization," *Appl. Artif. Intell.*, vol. 31, no. 4, pp. 299–315, 2017.
[38] V. Gonzalez-Huitron, J. A. León-Borges, A. E. Rodriguez-Mata, L. E. Amabilis-Sosa, B. Ramírez-Pereda, and H. Rodriguez, "Disease detection in tomato leaves via CNN with lightweight architectures implemented in raspberry Pi 4," *Comput. Electron. Agricult.*, vol. 181, Feb. 2021, Art. no. 105951.
[39] R. Thangaraj, S. Anandamurugan, and V. K. Kaliappan, "Automated tomato leaf disease classification using transfer learning-based deep convolution neural network," *J. Plant Diseases Protection*, vol. 128, no. 1, pp. 73–86, Feb. 2021.

EMRE ÖZBİLGE received the B.Sc. degree in computer engineering from Eastern Mediterranean University, North Cyprus, in 2006, the M.Sc. degree in intelligent systems from the University of Sussex, U.K., in 2007, the M.Sc. degree in computer science from the University of Essex, U.K., in 2008, and the Ph.D. degree in cognitive robotics from the University of Ulster, U.K., in 2013. He is currently a Lecturer at the Computer Engineering Department, Cyprus International University. His main research interests include deep learning, sequence learning, mobile robotics, novelty detection, machine learning, and machine vision.

MEHTAP KÖSE ULUKÖK received the B.S., M.S., and Ph.D. degrees in computer engineering from Eastern Mediterranean University, Famagusta, North Cyprus, in 1998, 2000, and 2009, respectively. She is currently an Assistant Professor at the Department of Computer Engineering, Bahçeşehir Cyprus University, Nicosia, North Cyprus. Her current research interests include machine learning, meta-heuristics, and cryptology.

ÖNSEN TOYGAR received the B.S., M.S., and Ph.D. degrees from the Computer Engineering Department, Eastern Mediterranean University, North Cyprus, in 1997, 1999, and 2004, respectively. Since September 2004, she has been working with the Computer Engineering Department, Eastern Mediterranean University. She worked as the Vice Chair of the Department, from September 2011 to January 2013. She is currently a Full Professor with the Department. Her current research interest includes biometrics, including face, iris and palm vein recognition, fusion of multimodal biometrics, computer vision, image processing, and plant disease recognition.

EBRU OZBİLGE received the B.S. and M.S. degrees from the Mathematics Department, Eastern Mediterranean University, North Cyprus, in 2000 and 2002, respectively, and the Ph.D. degree from the Mathematics Department, Kocaeli University, Turkey, in 2006. She had worked with the Department of Mathematics, Izmir University of Economics, from 2006 to 2016. Since August 2016, she has been working with the Department of Mathematics and Statistics, American University of the Middle East, Kuwait. She is currently a Full Professor with the Department and worked as the Department Chair. Her current research interests include inverse problems, fractional partial differential equations, and various numerical methods.