

# Tomato leaf disease classification by exploiting transfer learning and feature concatenation

Mehdhar S. A. M. Al-gaashani<sup>1</sup> | Fengjun Shang<sup>1</sup> | Mohammed S. A. Muthanna<sup>2</sup> |  
Mashaal Khayyat<sup>3</sup> | Ahmed A. Abd El-Latif<sup>4</sup> 

<sup>1</sup> College of Computer Science and Technology,  
Chongqing University of Posts and  
Telecommunications, Chongqing, China

<sup>2</sup> Department of Mathematics, Saint Petersburg  
State University, Saint Petersburg 199034, Russia

<sup>3</sup> Department of Information systems and  
Technology, College of Computer Science and  
Engineering University of Jeddah, Jeddah, Saudi  
Arabia

<sup>4</sup> Mathematics and Computer Science Department,  
Faculty of Science Menoufia University, Shebin  
El-Koom, Egypt

## Correspondence

Ahmed A. Abd El-Latif, Mathematics and Computer  
Science Department, Faculty of Science, Menoufia  
University, Shebin El-Koom, Egypt.  
Email: a.rahiem@gmail.com

## Funding information

National Nature Science Foundation of China,  
Grant/Award Number: 61672004

## Abstract

Tomato is one of the most important vegetables worldwide. It is considered a mainstay of many countries' economies. However, tomato crops are vulnerable to many diseases that lead to reducing or destroying production, and for this reason, early and accurate diagnosis of tomato diseases is very urgent. For this reason, many deep learning models have been developed to automate tomato leaf disease classification. Deep learning is far superior to traditional machine learning with loads of data, but traditional machine learning may outperform deep learning for limited training data. The authors propose a tomato leaf disease classification method by exploiting transfer learning and features concatenation. The authors extract features using pre-trained kernels (weights) from MobileNetV2 and NASNetMobile; then, they concatenate and reduce the dimensionality of these features using kernel principal component analysis. Following that, they feed these features into a conventional learning algorithm. The experimental results confirm the effectiveness of concatenated features for boosting the performance of classifiers. The authors have evaluated the three most popular traditional machine learning classifiers, random forest, support vector machine, and multinomial logistic regression; among them, multinomial logistic regression achieved the best performance with an average accuracy of 97%.

## 1 | INTRODUCTION

Tomato (*Solanum Lycopersicum* L.) is one of the most important and the most consumed crops worldwide. It is the second most important vegetable crop next to potato [1]. In the last four decades, the agricultural land area for tomato production increased by 164%, and the global tomato production was reached 182 million tons in 2018 [2]. China is by far the leading tomato producer in the world, producing 31% of the global tomato production.

Healthy tomato plants can be affected by various leaf diseases leading to reduced or poor-quality yields and, thus, can ultimately affect humans' health, livelihood, and economic security [3]. These leaf diseases are extremely damaging to fruit yield in tomato crops. Therefore, early

and accurate identification and classification of tomato leaf disease will help reduce yield losses and ensure Optimal production.

Traditionally, the classification of plant leaf diseases is conducted with a visual inspection of plant tissues by trained experts. A challenging problem that arises in this method is costly, time-consuming, and inefficient [4]. A number of works have shown that this problem can be overcome by automated plant disease recognition [5].

Recently, traditional machine learning (ML) algorithms have been widely applied in computer vision for automation plant disease classification [6]. The most common conventional ML approaches used for plant disease classification are neural networks [7], logistic regression, random forest (RF) [8], support vector machines (SVM) [9], adaptive boosting (AdaBoost)

[10], k-nearest neighbours (k-NN), naïve bayes [11]. However, these traditional machine learning techniques rely heavily on the features we provide them. The expert extracts these features manually, making these techniques expensive and time-consuming.

Automatic feature extraction using deep learning (DL) can easily tackle the limitation of the handcrafted features approach. DL is far superior to traditional ML, and it is widely used for plant disease classification [12]. A deep convolutional neural network (CNN) is the most adopted approach for plant disease classification [13]. Deep learning models depend upon large data and for training and demand a long time. However, traditional machine learning (e.g. logistic regression, random forest, or SVM) may outperform deep learning models for limited data training.

Furthermore, deep learning is more expensive when comparing to traditional machine learning approaches in terms of computational resources. For these reasons, the authors take advantage of both DL and ML to propose a method that exploits transfer learning techniques to extract features from deep lightweight CNN models and then feed them toward traditional ML classifiers. The proposed models are generally composed of four parts: the first part is lightweight pre-trained models applied as a fixed feature extractor, the second part is fusion features, the third part is dimensionality reduction using kernel PCA, the last part is training a shallow classifier. The authors evaluated three popular shallow classifiers SVM, RF, and MLR.

The main contribution of this study is as follows:

- I. The authors compare two state-of-the-art CNN pre-trained models MobileNetV2 and NASNetMobile, for image features used for plant disease classification.
- II. The authors proposed a Feature fusion of transfer learning features from MobileNetV2 and NASNetMobile models and perform dimensionality reduction of these fusion features using Kernel PCA. Afterwards, feature fusion is evaluated based on traditional ML classifiers SVM, RF, and MLR.

The remainder of this article is structured as follows:

Section 2 presents a literature review. Section 3 provides materials and methodology. Section 4 describes the proposed models. Section 5 illustrates the results of the proposed methods and compares these methods with other existing approaches. Lastly, the article is summarised in Section 6.

## 2 | RELATED WORK

Machine learning has been widely used in plant disease classification. Several methods are reported in the literature to automate plant disease classification. There are two general approaches: traditional machine learning approaches based on handcrafted features and deep learning approaches.

### 2.1 | Leaf disease recognition using traditional machine learning approaches

These approaches mainly rely on handcrafted features and typically follow a three-step process:

- Segmentation of disease and pest images utilizing well-known techniques such as colour conversion edge-based segmentation, mathematical morphology.
- Feature extraction from segmented images using texture, shape, size, and colour.
- Identification of plant diseases and pests using conventional machine learning algorithms

For instance, in [14], the authors proposed feature extraction techniques using Johnson SB distribution with scale-invariant feature transform (SIFT). The SIFT and colour statistic features were extracted by proposing methods; they feed them to multiclass support vector machine (Multi-Class SVM) classifiers for classification. The proposed methods achieved an accuracy score of 85.1% for tomato disease classification.

In [15], the authors proposed a novel approach based on the concatenation of multiple features. The local binary pattern features, Hu moments, Haralick, and colour histograms were extracted then concatenated. The extracted feature was fed to the decision tree and RF classifiers for tomato leaf disease classification. The proposed method achieved an accuracy score of 90% for the decision tree classifier and 94% for the random forest classifier.

A novel plant leaf classification method based on the combination of multiple descriptors is proposed by Kalyoncu et al. [16]. The shape, texture, geometric, and colour features are extracted from the photographic image of leaves. In addition, a new local binary pattern (LBP) variant, namely sorted uniform LBP, is also proposed for leaf texture description. After combining extracted features, the linear discriminant classifier (LDC) is applied for the classification phase. This method has been analysed on three datasets which are ICL, Flavia, Swedish and achieved an average accuracy score of 86.8%, 98.6% and 99.5% for ICL, Flavia, Swedish, respectively

In [17] the authors proposed a semi-automatic method for soybean leaf disease classification using colour and texture features and SVM as a classifier. The proposed method achieved an accuracy score of 90.00% using a big dataset of 4775 images.

### 2.2 | Leaf disease recognition using deep learning approaches

Researchers have focused on using deep learning methods for plant disease identification and diagnosis for the last few years. The following paragraphs show some of the current plant disease identification and diagnosis research based on deep learning methods.

In [18], a classification technique for tomato leaf disease has been proposed. The proposed technique used the

pre-trained model AlexNet [19] as feature extractor and kNN for classification diseases. The model achieved 76.1% accuracy.

Another study by [20] adopted a CNN model for tomato leaf disease classification based on transfer learning with fine-tuned MobileNetV2. The result showed that fine-tuning the MobileNetV2 model can identify the disease up to over 90% precision. In [21], the authors proposed a custom CNN model for tomato disease classification with an average accuracy of 91.2%.

In [22], an improved CNN model based on the Inception module and dilated convolution has been proposed for identifying 14 different crop diseases. The proposed CNN model achieved an accuracy score of 99.37% using PlantVillage dataset. In [23], the authors proposed a lightweight CNN model with eight hidden layers for identifying tomato plant disease images from the PlantVillage dataset and obtained an accuracy of 98.4%.

In [24], the authors proposed a nine-layer CNN model from scratch to identify 39 classes of plant leaves from the PlantVillage dataset. Various data augmentation approaches were used to increase the amount of training data. The proposed model achieved an average accuracy of 96.46%. Nithish et al. [25] proposed a transfer learning method for tomato leaf disease detection using a pre-trained ResNet-50 model. The ResNet-50 pre-trained model was fine-tuned to classify six classes and obtained an accuracy score of 97%.

### 3 | MATERIAL AND METHODOLOGY

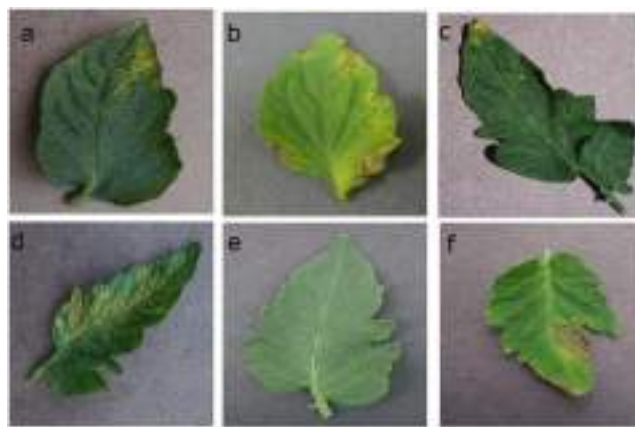
The MobileNetV2 and NASNetMobile pre-trained models employed for transfer learning in our approach are briefly addressed in this section, followed by a brief description of traditional machine learning classifiers.

#### 3.1 | Dataset description

The authors selected an images dataset from the public PlantVillage dataset [26]. The PlantVillage dataset was created by Hughes and Salathe [27], and it contains 54,309 labelled images for 14 different species, and 38 different classes comprise healthy and diseased foliage images. From the PlantVillage dataset, the authors used only 1152 images of tomato leaves divided into one healthy class and five unhealthy classes to extract features. These 1152 images were divided equally between classes (192 images per class).

The size of tomato leaf images was converted forcibly to  $224 \times 224$  before passing them as input for MobileNetV2 and NASNetMobile. The pixel values of the image dataset were normalised between 0 and 1.

Applying normalisation can speed up the process of transfer learning. For testing our model, the authors utilised a total of 288 images to validate our model (48 images for each class). In Figure 1, some samples from the PlantVillage dataset are shown.



**FIGURE 1** Leaf images from the PlantVillage dataset, one for each crop-disease that was utilised. (a) Bacterial spot, (b) yellow leaf curl virus, (c) Septoria leaf spot, (d) leaf mould, (e) healthy, (f) late blight

#### 3.2 | Transfer learning

Transfer learning is a machine learning technique that allows for adapting networks and models developed for certain specific applications to other applications [28]. For image classification, using deep neural network models from scratch is an expansive and time-consuming process. Furthermore, training deep learning models is highly relies on a large dataset which is challenging for a limited dataset to train [29]. To overcome these limitations, the authors have been used transfer learning of state-of-the-art pre-trained models. These pre-trained models can be utilised for prediction, feature extraction, and fine-tuning [30]. The authors have used lightweight pre-trained models MobileNetV2 and NASNetMobile as a fixed feature extractors. The key idea is to use these pre-trained models without their final classification layer to extract features and then use these features as input for shallow classifiers.

##### 3.2.1 | MobileNetV2

MobileNetV2 [32] is a CNN that is trained on greater than a million images from the ImageNet dataset. It was developed based on mobilenetv1 [33] that uses depthwise separable convolution to build a lightweight model. This technique can be split into two main parts depthwise convolution and pointwise convolution. The depthwise convolution executes filters across each input data channel individually. Output channels then convolve by pointwise convolution operation, but only with a  $1 \times 1$  filter. Depthwise separable convolution reduces the network's complexity cost and model size, which is suitable for mobile and embedded vision applications. In MobileNetV2, there are two types of blocks one is the bottleneck block with stride 1 and the other with stride 2. Figure 2 illustrates the main building block of MobileNetV2 architecture.

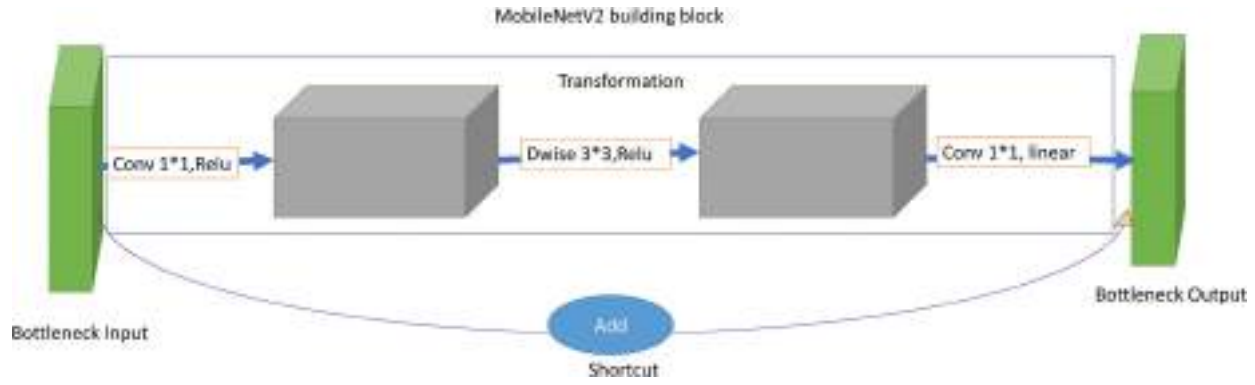


FIGURE 2 The main building block of MobileNetV2 architecture

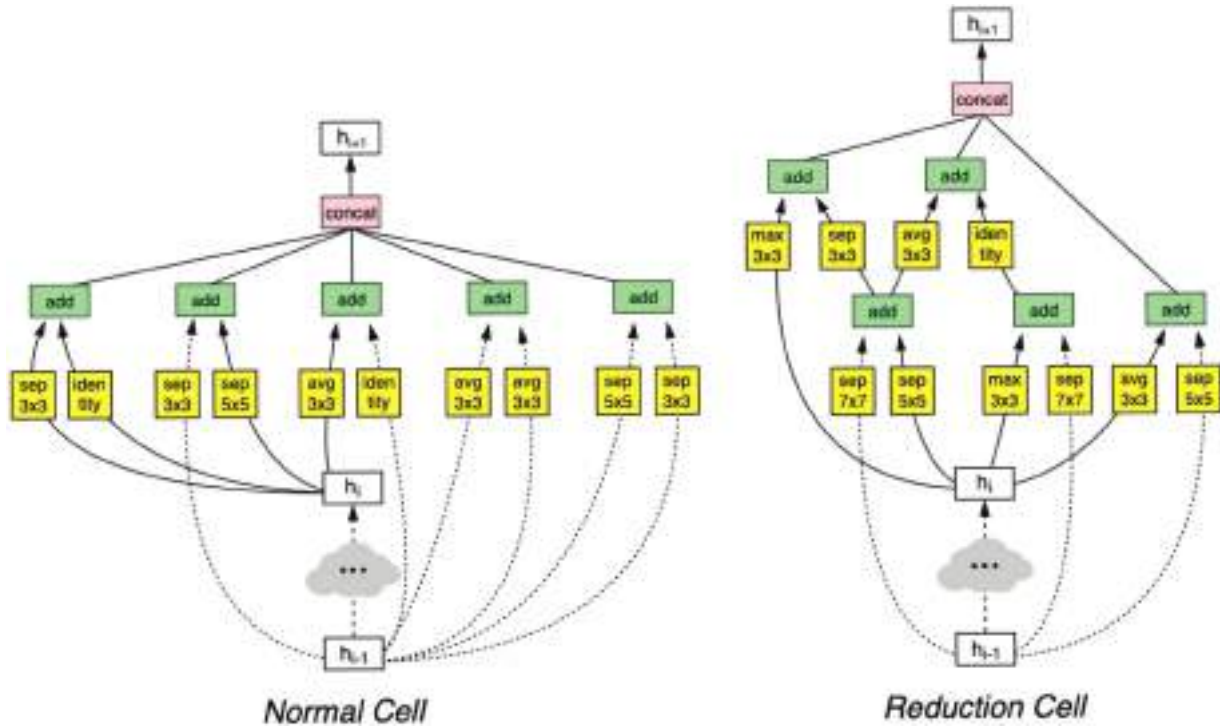


FIGURE 3 Normal cell and reduction cell architecture (Zoph et al., 2018)

### 3.2.2 | NASNetMobile

Neural architecture search Network (NASNet) is a promising scalable CNN architecture that automatically identifies architecture instead of hand-designed methods [29]. NASNet searched for optimal convolutional building blocks on a few training examples then transferred blocks to a big dataset. The NASNet architecture is composed of two types of cells called reduction cell and normal cell. The reduction cell will decrease the width and height of the input feature map size by half, where the normal cell will remain the same as the input feature map. The example of the appropriate structure of normal cell and reduction cell with five blocks is shown in Figure 3. NASNet model achieves superior results with smaller model size and lowers complexity compared to hand-designed architecture. NAS-

NetMobile is the mobile reduced version with 12 blocks (cells) and 5.3 million parameters. It has been pre-trained on the large ImageNet dataset [31]. Hence, the network has learned abundant feature representations for a broad set of images. This article used the NASNetMobile pre-trained model to perform transfer learning as a fixed feature extraction technique.

### 3.3 | Principal component analysis

Principal component analysis or PCA is a widely used method for linearity dimensionality reduction. It attempts to project original data from its high-dimension space to a lower dimension space while preserving the maximum variance of the data [34]. Kernel PCA exploits the idea of kernels to generalise PCA



to non-linearity cases [35]. The following steps can summarise the procedure of kernel PCA:

1. Choose a kernel function  $k(x, y)$  like the liner, RBF, Polynomial kernel and so forth.
2. Construct the kernel matrix/gram matrix  $K$  (covariance matrix)  $K_{i,j} = k(x(i), x(j))$
3. Centre the kernel matrix (the mean should be 0)
4. Calculate eigenvectors and corresponding eigenvalue of the centred kernel matrix, then order them such that they are decreasing.
5. The new reduced k dimension will be the k eigenvectors with the largest eigenvalues.

This study tested different kernels of PCA. The best performance was obtained by using a linear kernel.

### 3.4 | Random forest classifier

Random forest is a discriminative machine learning classifier for classification and regression purposes developed by [36]. It is a type of ensemble method called bootstrap aggregation or bagging. The philosophy behind bagging is to average noisy and unbiased to construct a model with low variance [37]. Given sample set to call it  $S$  (which is features vectors extracted from pre-trained models this study used), the random forest will create a lot of subsets  $S_1, S_2, S_3, \dots, S_m$  with random values (replace with sampling) then from this subsets will create multiple decision trees (classifiers) call them  $D_1, D_2, D_3, \dots, D_m$ . For these decision trees, a majority vote is taken for the predicted class of that tree.

A random forest is a powerful tool that can handle large data sets with higher dimensionality and still provides higher accuracy results. This study trained and applied RF to classify five tomato leaf disease classes and one healthy class.

### 3.5 | Multinomial logistic regression

Multinomial logistic regression (MLR) is a discriminative machine learning classifier that is an extension of logistic regression to the case when target  $y$  is a variable that ranges over more than two classes.

Suppose the  $x$  is the input vector feature to the MLR algorithm and  $k$  is the number of class labels (which is 6 in our study). Then using the softmax function, the MLR algorithm can compute the probability that  $x$  belonging to the class  $k$  by the formula:

$$p(y = k | x) = \frac{\exp(w_k \cdot x + b_k)}{\sum_{j=1}^k \exp(w_j \cdot x + b_j)} \quad (1)$$

where  $w_k, b_k$  are the parameters of the MLR algorithm that we want to optimise.

The loss function using for fitting the MLR algorithm is the cross-entropy loss which is the generalisation of binary cross-entropy loss of binomial logistic regression from 2 classes to  $k$  classes (also called negative log-likelihood loss) and can be written as follows:

$$\begin{aligned} L_{CE}(\hat{y}, y) &= -\log \hat{y}_k, (\text{where } k \text{ is the correct class}) \\ &= -\log \frac{\exp(w_k \cdot x + b_k)}{\sum_{j=1}^K \exp(w_j \cdot x + b_j)} \\ &\quad (\text{where } k \text{ is the correct class}) \end{aligned} \quad (2)$$

where  $y$  indicates the actual label and  $\hat{y}$  represents the predicted label.

As an optimiser for the loss function, a limited-memory BFGS was used (L-BFGS or LM-BFGS).

### 3.6 | Support vector machine

Support vector machine is a discriminative supervised machine learning algorithm that can be utilised for both classification and regression problems. However, it is mainly used for classification problems. It was developed by [38]. Given a set of training data, the objective of SVM is to mapping them to high dimensional space, then construct the best hyperplane or set of hyperplanes that separate data points to their potential classes. By using SVM kernel functions, the input data can be transformed to the required form. There are many kernels utilised by SVM like linear, polynomial, radial basis function and sigmoid and so forth. This work utilised the linear kernel as it showed the best performance compared to the rest. The linear kernel is faster and required fewer parameters to optimise (only the  $C$  regularisation parameter). The linear kernel can be written as follow:

$$k(x_i, x_j) = x_i \cdot x_j \quad (3)$$

where  $x_i, x_j$  denote the input features.

By maximising the margin from the support vectors to the hyperplane, the best separation between data points can be achieved. To find the maximum margin, the SVMs aims to tackle the following optimisation problem:

$$\begin{aligned} \min_{\omega, b, \zeta} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^n \zeta_i \\ & y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i \end{aligned} \quad (4)$$

where  $\zeta$  denotes the distance to correct margin with  $\zeta \geq 0$ ,  $i = 1, \dots, n$

where  $C$  denotes a regularisation parameter

where  $w^1 w = \|w^2\|$  denotes the normal vector

where  $\phi(x_i)$  denotes the transformed input vector

where  $b$  denotes a bias parameter

where  $y_i$  denotes the  $i_{th}$  target value

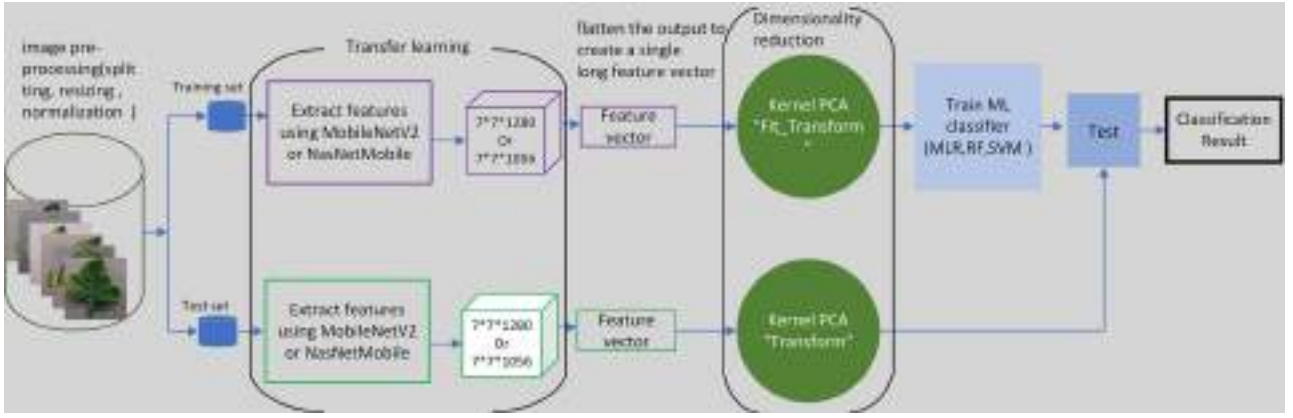


FIGURE 4 Architecture of a simple transfer learning method

## 4 | PROPOSED MODEL

### 4.1 | Proposal 1 - simple transfer learning approach

The first proposal seeks to estimate in the simplest manner the effectiveness of two lightweight CNN architectures pre-trained on the ImageNet database to extract features that are relevant to the classification of tomato disease images. The architectures evaluated are MobileNetV2 or NASNetMobile. After pre-processing, the images were resized to meet the requirement of the dimension of the input layer MobileNetV2 or NASNetMobile model (that is the same shape  $224 \times 224$  for both models) and then propagated in the network.

The features are extracted from the top layer of each CNN model individually before the classification part. These features were flattened into a one-row feature vector, which we will refer to as  $V_{Mobnet}$  for those extracted using the Mobilenetv2 model and  $V_{Nasnet}$  for those extracted using the NASNetMobile model. The  $V_{Mobnet}$  vector contains 62720 features, and the  $V_{Nasnet}$  vector has 51744 features. These feature vectors were reduced using the kernel PCA technique. Three PCA kernels, linear, residual, and RBF, were evaluated.

The extracted and reduced feature vectors were then used as input for training a shallow classifier. The SVM, RF, and MLR classifiers were evaluated for each input feature vector. For parameter optimisation of the classifiers, the authors used a grid search method. Figure 4 demonstrates all steps of this proposal.

### 4.2 | Proposal 2- Concatenated based approach

The second proposal aims to boost classification by concatenation features extracted from different pre-trained models. After preprocessing, the images were resized to the input shape of the MobileNetV2 and NASNetMobile then in parallel propagated in the networks.

The same feature vectors were obtained as in the first proposal, which is the  $V_{Mobnet}$  feature vector for the MobileNetV2 model and the  $V_{Nasnet}$  feature vector for NASNetMobile. Once the feature vectors had been flattened, they were then concatenated to create a single feature vector. The concatenation of the two feature vectors to form a hybrid feature vector is mathematically expressed as:

$$V = [V_{Mobnet}, V_{Nasnet}] \quad (5)$$

The single feature vector  $V$  then reduced using kernel PCA. Three PCA kernels, linear, residual, and RBF, were evaluated. The compressed single feature vector was then utilised as the input for a conventional machine learning classifier. Three ML classifiers were evaluated that is SVM, RF, and MLR. To find the best hyperparameters for the classifiers, a grid search approach is utilised. Figure 5 illustrates all steps of this proposal.

## 5 | RESULT AND DISCUSSION

### 5.1 | Performance metrics

The evaluation results of the proposed model are presented in Figure 10, which shows confusion matrices of SVM, RF, and MLR classifiers using concatenated features with and without applying kernel PCA.

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \quad (6)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

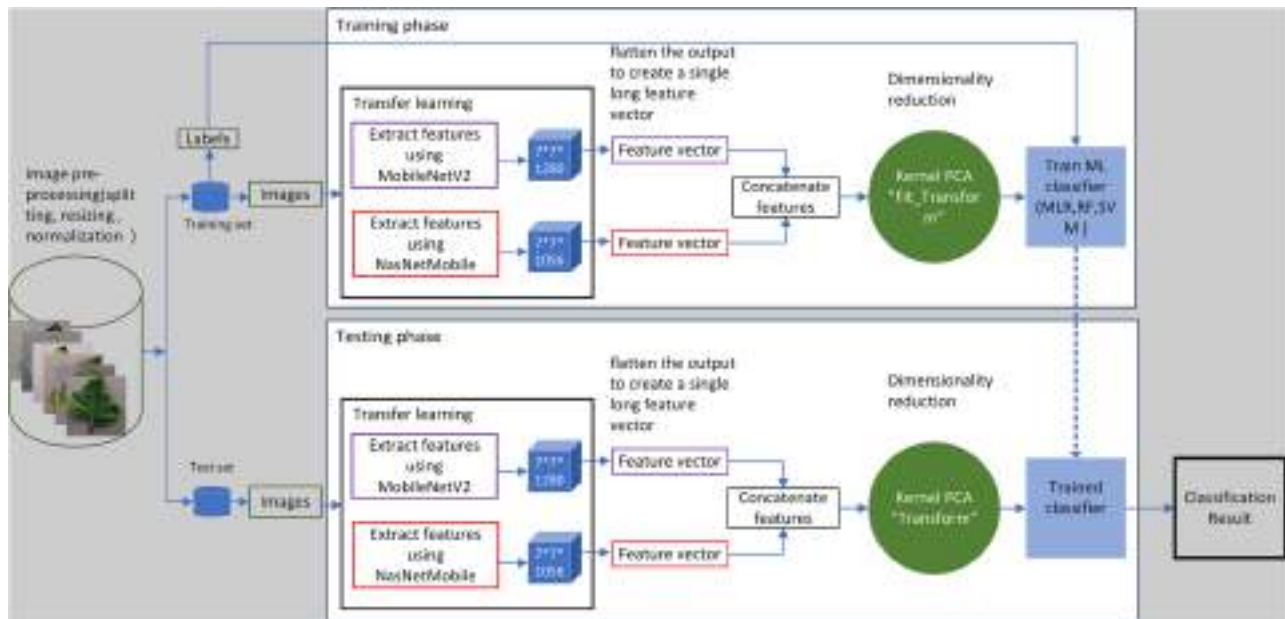


FIGURE 5 Architecture of concatenation-based method



FIGURE 6 Computation time analysis of the proposed method

In these equations, “TP” denotes true positive, “TN” means true negative, “FP” points to false positive, and “FN” describes false negative. Tables 2–4 show the details of our result. Figures 7–9 show the accuracy of SVM, RF, and MLR for all proposed models

## 5.2 | Result evaluation and analysis

All simulations were performed using Python programming language version 3.7.4 for windows with the Keras v2.3.1 framework [39]. Google’s library tensorflow v2.0.0, was backend to Keras. The SVM, RF, and MLR classifiers were applied using the open-source “scikit-learn” module [40]. The hardware setup was: CPU i7-3537 2.50 GHz and 8 GB RAM.

At first, the MobileNetV2 model with pre-trained filters of ImageNet is applied to extract features from tomato leaf images. Following that, we reduced these features using various kernel principal component analyses (kernel PCA), and then utilised

## PERFORMANCE ANALYSIS OF DIFFERENT CLASSIFIERS- NASNETMOBILE

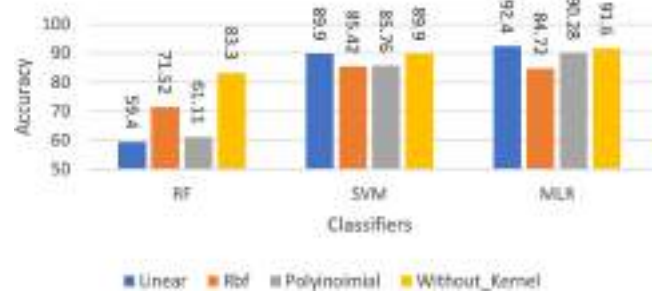


FIGURE 7 Performance analysis of RF, SVM, and MLR with and without applying KernelPCA when the NASNetMobile features are used as input for classifiers

them as input to a shallow classifier for training. Experimental results indicated that using the linear kernel yielded improved accuracy compared to the RBF and polynomial kernels, and the resulting accuracy beat both the RBF and the polynomial kernels. In contrast, the performance of the RF classifier is decreased when applying KernelPCA. On the other hand, the performance of the SVM classifier with linear kernel showed the same performance as SVM without KernelPCA. The overall accuracy, in this case, is 88.2% for RF, 93.8% for SVM, and 93.8% for MLR. The same approach has been applied with NASNetMobile model and a less efficient result has been achieved than MobileNetV2 model with overall accuracy, 59.4% for RF, 89.9% for SVM, and 92.4% for MLR.

**TABLE 1** Comparison with other existing techniques

Related work	Features	Model	Dataset	Number of classes	Number of training data	Original classification accuracy (%)	Classification accuracy on our dataset with six classes (%)
Hlaing et al. [14]	SIFT Texture feature + statistical colour features	SVM	PlantVillage	7	3,535	85	—
Zaki et al. [20]	Fine-tune MobileNetV2	MobileNetV2	PlantVillage	4	3,471	95.6	90.3
Basavaiah et al. [15]	Hu Moments, Local Binary Pattern features, and colour histograms.	Random forest and decision tree	PlantVillage	5	300	90 for decision tree and 94 for RF	—
Agarwal et al. [21]	Model-based	Typical CNN model	PlantVillage	10	10,000	91.2	87
Nithish et al. [25]	Model-based	ResNet-50	PlantVillage	6	12,206	97	81.4
<b>Proposed method 1</b>	<b>Fixed feature extractor of MobileNetV2 or NASNetMobile</b>	<b>Multinomial logistic regression (MLR)</b>	PlantVillage	<b>6</b>	<b>1,152</b>	<b>93.8 for MobileNetV2 and 92.4 for NASNetMobile</b>	<b>93.8 for MobileNetV2 and 92.4 for NASNetMobile</b>
<b>Proposed method 2</b>	<b>Combination features of MobileNetV2 + NASNetMobile</b>	<b>Multinomial logistic regression (MLR)</b>	PlantVillage	<b>6</b>	<b>1,152</b>	<b>97</b>	<b>97</b>

**TABLE 2** The Precision, Recall, and F1 score based on the concatenation method with an SVM classifier

Classes	Precision	Recall	F <sub>1</sub> -score
Bacterial spot	95	98	91
Late blight	93	83	88
Leaf mould	100	100	100
Septoria spot	98	100	99
Yellow curl virus	96	98	97
Healthy	100	92	96

**TABLE 3** The Precision, Recall, and F1 score based on the concatenation method with an RF classifier

Classes	Precision	Recall	F <sub>1</sub> -score
Bacterial spot	75	83	79
Late blight	71	71	73
Leaf mould	96	94	95
Septoria spot	89	98	93
Yellow curl virus	89	81	85
Healthy	86	77	81

Next, the proposed concatenation-based approach is implemented by using concatenated features to training our classifiers. The accuracy obtained of all classifiers with and without applying kernel PCA is presented in Figure 9. Also, the result indicates that the performance of RF is degraded when used kernel PCA, whereas SVM and MLR are improved. Moreover, the filters of MobileNetV2 showed a better ability to extract relevant features than the NASNetMobile model.

The learning performance of different classifiers may be influenced by their hyperparameters, which were improved using a grid search approach in our proposed model.

The number of decision trees in the random forest is set to 400, and the depth of each decision tree is set to 70; in the support vector machine (SVM), the penalty parameter C is set to 0.1, the gamma parameter is set to 0.001, and the linear kernel is

utilised as the kernel. We set the C parameter to 0.1, the penalty to 12, and utilised the 'lbfgs' solver as an optimiser for multinomial logistic regression.

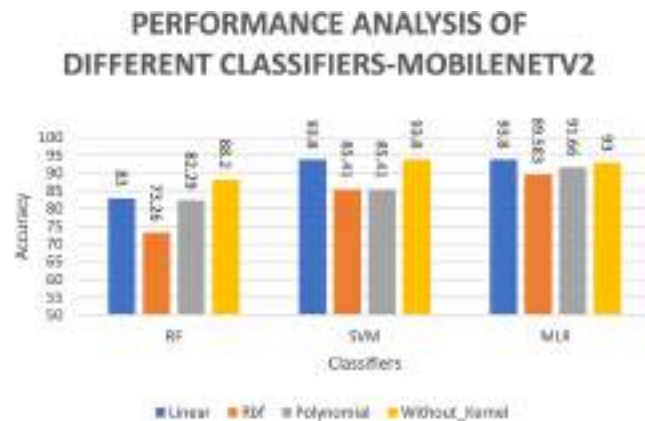
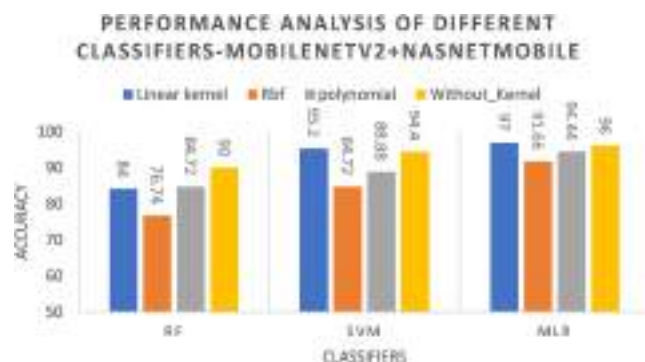
F1 score, Recall, and Precision of the concatenation-based approach are presented in Tables 2–4 for RF, SVM, and MLR classifiers, respectively. The concatenation-based model achieved the best performance result. All the evaluating parameters improved noticeably after the implementation of the proposed concatenation method. Figures 7–9 demonstrate the performance analysis of RF, SVM, and MLR classifiers before and after using the KernelPCA method. The figures show that the concatenation-based strategy outperforms the single feature extraction approach.

To validate our result, the results were compared with other state-of-art approaches on tomato leaf disease classification.



**TABLE 4** The Precision, Recall, and F1 score based on the concatenation method with an MLR classifier

Classes	Precision	Recall	F <sub>1</sub> -score
Bacterial spot	94	98	96
Late blight	92	96	99
Leaf mould	99	96	100
Septoria spot	100	100	100
Yellow curl virus	94	98	96
Healthy	100	96	98

**FIGURE 8** Performance Analysis of RF, SVM, and MLR with and without applying KernelPCA when the MobileNetV2 features are used as input for classifiers**FIGURE 9** Performance analysis of RF, SVM, and MLR with and without applying KernelPCA when the MobileNetV2 features are used as input for classifiers

The proposed model, which was constructed by concatenating the features of MobileNetV2 and NASNetMobile, demonstrated superior performance in classifying tomato leaf diseases. Table 1 compares the results of this investigation to those of earlier studies that classified tomato leaf disease using computer vision. The results indicate that our method outperforms several existing state-of-the-art techniques for tomato leaf disease classification. However, when the mean classification accuracy is considered, this study's classification accuracy is comparable to that of several other research. In contrast to previous studies,

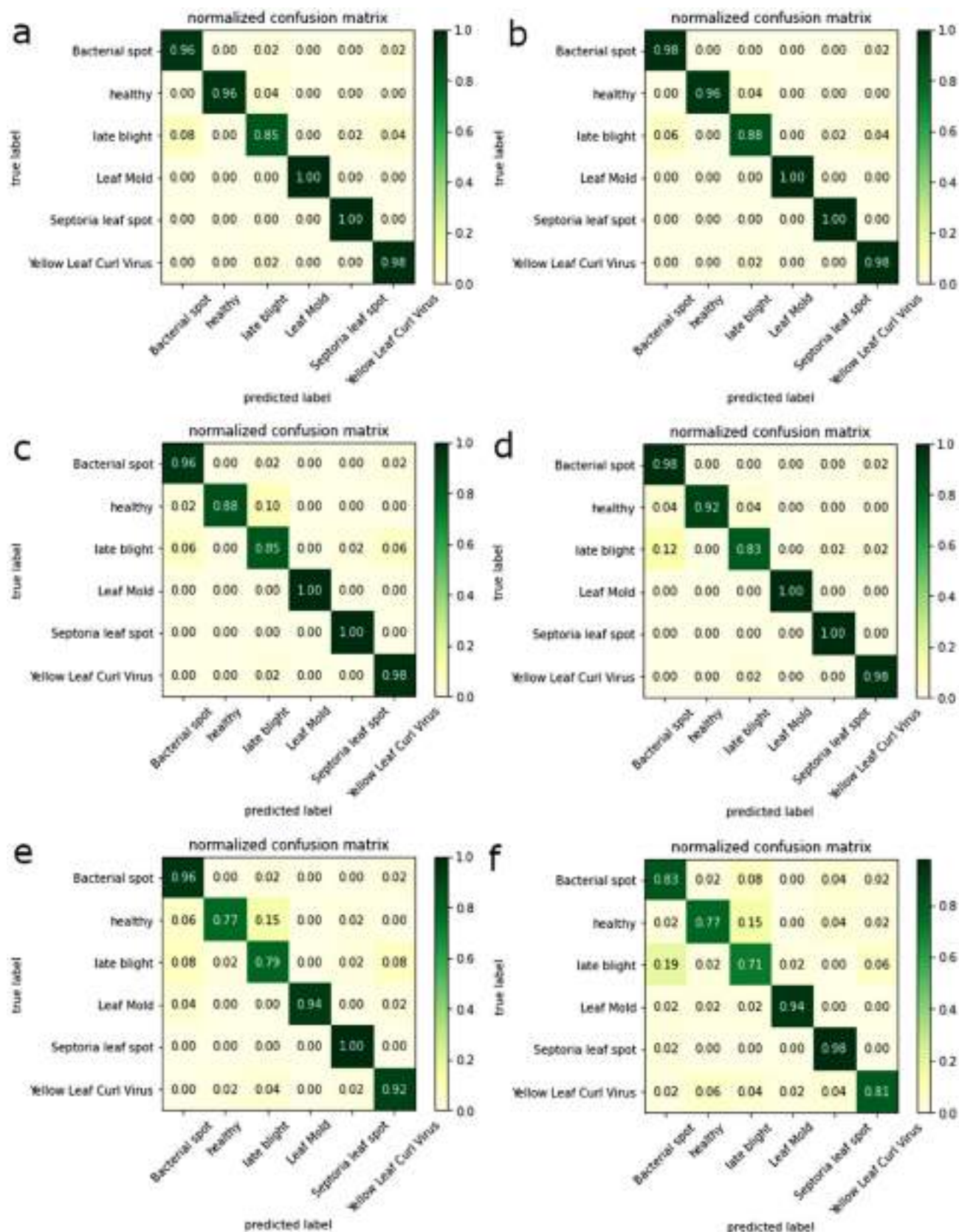
the number of images this study used is significantly less than in previous studies. At the same time, the training, testing, and feature extraction times are significantly reduced compared to earlier methods.

It is, however, impossible to make a direct comparison between all of these approaches. The traditional methods used by Hlaing et al. [14] and Basavaiah et al. [15] rely on handcrafted features. These handcrafted features are not robust, expensive, and time-consuming. Hence our proposed method has automated the feature extraction process and reduced the time extraction. In contrast, the deep learning methods used by Zaki et al. [20], Agarwal et al. [21] and Nithish et al. [25] showed classification accuracy comparable to our methods.

However, the training and testing time of our proposed method is much lower than those studies. In addition, the number of images this study has, has been reduced significantly. The deep learning methods described by Zaki et al. [20], Agarwal et al. [21] and Nithish et al. [25] have been implemented and tested against our method using the same dataset (our dataset with six classes). For instance, Agarwal et al.'s [21] typical CNN model, which consists of three convolutional layers, three max-pooling layers, and two fully connected layers, was trained and tested on a dataset containing 10,000 images and achieved an average accuracy of 87%. However, when we trained and tested this model on our dataset (1152 images), we observed that its accuracy decreased to 87%. Additionally, Nithish et al. [25] reported that ResNet-50 was trained on a dataset containing 12206 images; however, when we trained and tested it on our dataset (which contained only 1152 images), the accuracy declined dramatically from 97% to 81.4%. It demonstrates that a small dataset can result in suboptimal performance when deep learning models with large parameters, such as ResNet-50 (25 million parameters), are used.

Furthermore, one of the most widely used metrics for assessing the performance of machine learning techniques is the area under the receiver operating characteristic (AUC–ROC) curve. ROC curves are frequently used in binary classification to analyse a classifier's output. However, by utilising the One vs. All approach, we have extended it to multiclass classification problems. This technique is based on fitting  $K$  one-vs-all classifiers, with group  $s_i$  set as the positive class in the  $i$ -th iteration and all classes  $s_j$  with  $j \neq i$  set as the negative class. The AUC–ROC curves for five randomly selected classes from the testing dataset are shown in Appendix B. As seen in Appendix B, it is beneficial to perform dimensionality reduction before SVM and MLR since they perform better in the reduced space.

The AUC–ROC curves for five randomly selected classes from the testing dataset are shown in Appendix B. As seen in Appendix B, it is beneficial to perform dimensionality reduction before SVM and MLR since they perform better in the reduced space. In contrast, the performance of RF has been decreased after applying Kernel PCA since the random forest is essentially bootstrap resampling and training decision trees on samples, which makes it not a cure for small training data. Additionally, the MobileNetV2 model outperformed the NASNetMobile model in terms of feature extraction time and ability to extract relevant features, as illustrated in Figure 6. The next



**FIGURE 10** Confusion matrices of RF, SVM, and MLR based on concatenated features with and without kernel PCA: (a) MLR, (b) MLR with KernelPCA, (c) SVM, (d) SVM with kernelPCA, (e) RF, (f) RF with KernelPCA

step was to assess the practical elements of implementing the algorithm to determine its real-world usefulness. The training time and testing time of each classifier before and after applying the Kernel PCA are shown in Figure 6. As seen in Figure 6, The training time and testing time of each classifier have been drastically reduced after applying the KernelPCA method.

## 6 | CONCLUSION AND FUTURE WORK

In our paper, two methods for the classification of tomato leaf disease were used. The features from the images were extracted individually from each of the pre-trained models in the first experimental method. After extraction, these features were reduced using Kernel PCA then passed to a shallow classifier for the classification. Three popular classifiers were assessed that is SVM, RF, and MLR. These classifiers were tested using 20% of our dataset. In the second method, we followed the same previous steps as in the first method. Instead of using features individually from each CNNs pre-trained model, we combine these features into one feature vector to boost accuracy and come closer to the best classification practices. In comparison with other state-of-art techniques, our proposed method achieved comparable results. Therefore, our model described above can be utilised by farmers to diagnose diseases occurring in the leaves of tomato plants.

In our study, the types of plant leaf disease are limited to tomato bacterial spot, tomato healthy, tomato late blight leaf, tomato mould Septoria, tomato leaf spot, and tomato yellow leaf curl virus due to the fact that they are the most common sorts. The proposed architectures can be developed in the future to identify new types of plant leaf disease. Furthermore, one of the major disadvantages of the proposed recognition system is that the images utilised in our experiment were obtained in laboratory conditions; nevertheless, our approach might be further enhanced to support an integrated plant disease detection system that operates in real-world settings.

In the future, we will explore and apply new transfer learning techniques with multi-layer feature extraction on pre-trained models to select the best layers that provide the optimal features. Furthermore, we plan to explore ensemble methods (combine of classifiers outputs) to boosting prediction results.

## ACKNOWLEDGEMENTS

The authors would like to thank the National Nature Science Foundation of China (No. 61672004).

## CONFLICT OF INTEREST

The authors do not have financial and personal relationships with other people or organizations that could inappropriately influence (bias) their work.

## DATA AVAILABILITY STATEMENT

Dataset is available from its original source as cited in the article. The PlantVillage dataset is available at <https://github.com/spMohanty/PlantVillage-Dataset>

## ORCID

Abmed A. Abd El-Latif  <https://orcid.org/0000-0002-5068-2033>

## REFERENCES

- Home Food and Agriculture Organization of the United Nations. <http://www.fao.org/faostat/en/#home> (2019). Accessed 1 February 2021
- Griencisen, M.L., Aegerter, B.J., Scott Stoddard, C., Zhang, M.: Yield and fruit quality of grafted tomatoes, and their potential for soil fumigant use reduction. A meta-analysis. *Agron. Sustain. Dev.* 38, 29 (2018). <https://doi.org/10.1007/s13593-018-0507-5>
- Savary, S., Ficke, A., Aubertot, J.N., Hollier, C.: Crop losses due to diseases and their implications for global food production losses and food security. *Food Security* 4, 519–537 (2012)
- Islam, M., Dinh, A., Wahid, K., Bhowmik, P.: Detection of potato diseases using image segmentation and multiclass support vector machine. *Can. Conf. Electr. Comput. Eng.* 39–42 (2017)
- Azlah, M.A.F., Chua, L.S., Rahmad, F.R., Abdullah, F.I., Alwi, S.R.W.: Review on techniques for plant leaf classification and recognition. *Computers* 8(4), 77 (2019)
- Ngugi, L.C., Abelwahab, M., Abo-Zahhad, M.: Recent advances in image processing techniques for automated leaf pest and disease recognition – A review. *Information Processing in Agriculture* 8 (2020)
- Ramya, V., Lydia, M.A.: Leaf Disease Detection and Classification using Neural Networks. *Int. J. Adv. Res. Comput. Commun. Eng.* (2016)
- Sujatha, R., Chatterjee, J.M., Jhanjhi, N.Z., Brohi, S.N.: Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocess. Microsyst.* 80, 103615 (2021)
- Rumpf, T., Mahlein, A.K., Steiner, U., Oerke, E.C., Dehne, H.W., Plümer, L.: Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. *Comput. Electron. Agric.* 74(1), 91–99 (2010)
- Kantale, P.: Pomegranate disease classification using Ada-Boost ensemble algorithm. *Int. J. Eng. Research & Technology* 9(09), 612–620 (2020)
- Johannes, A., Picon, A., Alvarez-Gila, A., et al.: Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. *Comput. Electron. Agric.* 138, 200–209 (2017)
- Hasan, R.I., Yusuf, S.M., Alzubaidi, L.: Review of the state of the art of deep learning for plant diseases: A broad analysis and discussion. *Plants* 9(10), 1–25 (2020)
- Hong, H., Lin, J., Huang, F.: Tomato Disease Detection and Classification by Deep Learning. In: *Proc. - 2020 Int. Conf. Big Data, Artif. Intell. Internet Things Eng. ICBAIE 2020*, pp. 25–29 (2020)
- Hlaing, C.S., Zaw, S.M.M.: Tomato Plant Diseases Classification Using Statistical Texture Feature and Color Feature. In: *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, pp. 439–444 (2018)
- Basavaiah, J., Arlene Anthony, A.: Tomato leaf disease classification using multiple feature extraction techniques. *Wirel. Pers. Commun.* 115(1), 633–651 (2020)
- Kalyoncu, C., Toygar, Ö.: GTCLC: Leaf classification method using multiple descriptors. *IET Comput. Vis.* 10(7), 700–708 (2016)
- Kaur, S., Pandey, S., Goel, S.: Semi-automatic leaf disease detection and classification system for soybean culture. *IET Image Process.* 12(6), 1038–1048 (2018)
- Batool, A., Hyder, S.B., Rahim, A., Waheed, N., Asghar, M.A.: Fawad: Classification and identification of tomato leaf disease using deep neural network. In: *2020 Int. Conf. Eng. Emerg. Technol. ICEET 2020* (2020)
- Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*. (2012)
- Zaki, S.Z.M., Zulkifley, M.A., Mohd Stofa, M., Kamari, N.A.M., Mohamed, N.A.: Classification of tomato leaf diseases using mobilenet v2. *IAES Int. J. Artif. Intell.* (2020)



21. Agarwal, M., Singh, A., Arjaria, S., Sinha, A., Gupta, S.: ToLeD: Tomato leaf disease detection using convolution neural network. In: *Procedia Computer Science*. (2020)
22. Wang, L., Sun, J., Wu, X., Shen, J., Lu, B., Tan, W.: Identification of crop diseases using improved convolutional neural networks. *IET Comput. Vis.* 14(7), 538–545 (2020)
23. Agarwal, M., Gupta, S.K., Biswas, K.K.: Development of Efficient CNN model for Tomato crop disease identification. *Sustain. Comput. Informatics Syst.* 28, 100407 (2020)
24. Geetharamani, G., Arun, P.J.: Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput. Electr. Eng.* 76, 323–338 (2019)
25. Nithish, E.K., Kaushik, M., Prakash, P., Ajay, R., Veni, S.: Tomato leaf disease detection using convolutional neural network with data augmentation. In: *Proc. 5th Int. Conf. Commun. Electron. Syst. ICCES 2020*, pp. 1125–1132 (2020)
26. Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. *Front. Plant Sci.* 7(September), 1–10 (2016)
27. Hughes, D.P., Salathé, M.: An open access repository of images on plant health to enable the development of mobile disease diagnostics. *Computers and Society* (2015)
28. Yang, Q., Zhang, Y., Dai, W., Pan, S.J.: *Transfer Learning* (2020)
29. Xu, W., Yu, G., Zare, A., et al.: Overcoming small minirhizotron datasets using transfer learning. *Comput. Electron. Agric.* 175(April), 105466 (2020)
30. Kaya, A., Keceli, A.S., Catal, C., Yalic, H.Y., Temucin, H., Tekinerdogan, B.: Analysis of transfer learning for deep neural network based plant classification models. *Comput. Electron. Agric.* 158(January), 20–29 (2019)
31. Russakovsky, O., Deng, J., Su, H., et al.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115(3), 211–252 (2015)
32. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520 (2018)
33. Howard, A.G., Zhu, M., Chen, B., et al.: MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv*, (2017)
34. Bishop, C.M.: *Pattern Recognition and Machine Learning*. (2006)
35. Wang, Q.: *Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models* (2012)
36. Breiman, L.: Random forests. *Mach. Learn.* (2001)
37. Yaman, E., Subasi, A.: Comparison of bagging and boosting ensemble machine learning methods for automated EMG signal classification. *Biomed Res. Int.* 2019, 1–13 (2019)
38. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* 20, 273–297 (1995)
39. keras-team/keras: Deep Learning for humans. <https://github.com/fchollet/keras> (2015). Accessed March 2021
40. Pedregosa, F., Varoquaux, G., Gramfort, A., et al.: Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* (2011)

**How to cite this article:** Al-gaashani, M.S.A.M., Shang, F., Muthanna, M.S.A., Khayyat, M., Abd El-Latif, A.A.: Tomato leaf disease classification by exploiting transfer learning and feature concatenation. *IET Image Process.* 16, 913–925 (2022).  
<https://doi.org/10.1049/ipr2.12397>

## APPENDIX A: SUPPLEMENTARY CODE

Code related to this study is available at GitHub link: <https://github.com/mehdhar/Tomato-disease-classification>

## APPENDIX B: THE AUC–ROC CURVES OF THE PROPOSED MODEL

