electronics

MDPI

Article

A New Compact Method Based on a Convolutional Neural Network for Classification and Validation of Tomato Plant Disease

Shivali Amit Wagle [1], Harikrishnan R [1,*], Vijayakumar Varadarajan [2,3,4,*] and Ketan Kotecha [5]

[1] Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, India
[2] School of NUOVOS, Ajeenkya DY Patil University, Pune 412105, India
[3] School of Computer Science and Engineering, UNSW, Sydney 2052, Australia
[4] Swiss School of Business and Management, SSBM, 1213 Geneva, Switzerland
[5] Symbiosis Centre for Applied Artificial Intelligence (SCAAI), Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, India
* Correspondence: dr.rhareish@gmail.com (H.R.); vijayakumar.varadarajan@gmail.com (V.V.)

Abstract: With recent advancements in the classification methods of various domains, deep learning has shown remarkable results over traditional neural networks. A compact convolutional neural network (CNN) model with reduced computational complexity that performs equally well compared to the pretrained ResNet-101 model was developed. This three-layer CNN model was developed for plant leaf classification in this work. The classification of disease in tomato plant leaf images of the healthy and disease classes from the PlantVillage (PV) database is discussed in this work. Further, it supports validating the models with the images taken at "Krishi Vigyan Kendra Narayangaon (KVKN)," Pune, India. The disease categories were chosen based on their prevalence in Indian states. The proposed approach presents a performance improvement concerning other state-of-the-art methods; it achieved classification accuracies of 99.13%, 99.51%, and 99.40% with N1, N2, and N3 models, respectively, on the PV dataset. Experimental results demonstrate the validity of the proposed approach under complex background conditions. For the images captured at KVKN for predicting tomato plant leaf disease, the validation accuracy was 100% for the N1 model, 98.44% for the N2 model, and 96% for the N3 model. The training time for the developed N2 model was reduced by 89% compared to the ResNet-101 model. The models developed are smaller, more efficient, and less time-complex. The performance of the developed model will help us to take a significant step towards managing the infected plants. This will help farmers and contribute to sustainable agriculture.

Keywords: artificial intelligence; classification; prediction; ResNet-101; validation

check for updates

Citation: Wagle, S.A.; R, H.; Varadarajan, V.; Kotecha, K. A New Compact Method Based on a Convolutional Neural Network for Classification and Validation of Tomato Plant Disease. *Electronics* **2022**, *11*, 2994. https://doi.org/10.3390/electronics11192994

Academic Editors: Jungong Han, Guiguang Ding and Luca Mesin

Received: 18 July 2022
Accepted: 15 September 2022
Published: 21 September 2022

1. Introduction

As the world's population grows, so does the demand for healthy, high-quality food. Agriculture is one of the most significant factors in the economy for many countries. It is considered a way of life and a national focus. Farming enables people with little or no farming experience to grow plants or crops [1]. "The farmers should take preventive steps to protect the farm from diseases that can be proactively prevented if the cause of the disease is known in advance. Traditional techniques used are cumbersome and expensive" [2,3]. "The diagnosis of diseases, if mistaken by the experts due to the sizeable cultivating area they have to inspect and treating the plants, may not be sufficient to save the plant or reduce the diseases in them" [4]. "As a part of the concern, the farmers followed the steps to spray pesticides or chemical fertilizers to get rid of the diseases." However, this harms the crop along with the ecosystem.

The multidisciplinary approach incorporates botanical data, the species concept, and computer-aided plant classification solutions [5]. Botanists can now use computer vision approaches to help them identify plants thanks to advances in science and technology. Computer vision researchers used plant leaves as a comparative tool to classify plants [6]. "The introduction to the deep learning model techniques in the field of classification and detection" was introduced by [7] with the essential deep-learning tool that is the CNN. Recent advances in deep learning, particularly in convolutional neural networks (CNNs), have resulted in significant breakthroughs in a variety of applications, including the classification of plant diseases [8]. Robust and cost-efficient schedules for decision-making in multimode projects were discussed by [9]. The investigation of twenty artificial intelligence techniques was discussed by [10] for decision-making.

AlexNet, GoogLeNet, ResNet-50, ResNet-18, ResNet-101, VGG 16, VGG 19, DenseNet, SqueezeNet, and other pre-trained CNN models differ in terms of layer depth and nonlinear functions used in them. He et al. [11] "conferred a residual network (ResNet) that skips connections between ReLU and normalization layers. ResNet has many residual blocks; this helps proficiently with the deeper learning models. Transfer learning can use the pre-trained network and modify some parts according to work needs" [12,13]. The model has the same structure, with four essential layers: "convolution layer," "pooling layer," "fully-connected layer," and "output layer." The model by Hu et al. [14] is a less complex algorithm that achieves a precision of 94.26%. Bandwidth usage increases when one tries to transfer data from sensor to server. The processing time decides on the network's bandwidth utilization [15]. A model that offers improvements in computational speed and model size helps in reducing bandwidth utilization [16].

Deep learning methods have demonstrated significant improvements in plant leaf classification performance [17]. "Data augmentation influences the average precision of the class" [18,19]. A new model was developed by [20] for fault detection using k-means clustering for risk management decision making. At every step, k-means algorithm moves each mean value to the center of the next step and then updates by recomputing the distance between each failure mode and its nearest central vectors. The steps will be repeated till the cluster results of these two iterations no longer vary. After that, the cluster converges, and the final k clusters are formed for decision making. Hasan et al. [21] reviewed the most current training, dataset training, data augmentation, feature extraction, crop recognition, plant enumeration, and plant disease detection techniques, and the performances of classifiers. Liu et al. [22] developed "a ten-layer CNN model and got an accuracy of 87.92% with the Flavia dataset." Mukti et al. [23] "used transfer learning for the deep learning models AlexNet, VGG 16, VGG 19, and ResNet-50 in their work to classify images of plant leaves, achieving an accuracy of 99.80% with the ResNet-50 mode." Classification of nine plant species using AlexNet and support vector machine was done by [24] and achieved an accuracy of 91.15%. Jadhav et al. [25] "classified soybean disease with the deep learning networks of AlexNet, GoogLeNet, VGG 16, ResNet-101, and DenseNet 201. GoogLeNet and VGG 16 had the highest accuracy of 96.4% compared to the other networks." Chen et al. [26] achieved an accuracy level of 84.25% with the INC-VGGN model on the PV dataset for the classification of rice plant disease, and the model's performance was improved to 91.83% on their dataset. The classification of four paddy leaf diseases by [27] ResNet-101 achieved 91.52% accuracy . The dataset of paddy leaves consisting of brown spots, leaf blast, leaf blight, leaf smut, and a healthy class was collected from Kaggle and UCI repository. A Faster R-CNN algorithm was used by [28] to diagnose rice plant disease and attained an average accuracy of 98.84% for healthy and three disease classes. Rangarajan et al. [29] used "six different deep learning models, viz., AlexNet, VGG16, VGG19, GoogLeNet, ResNet101, and DenseNet201, in classifying ten classes of four varieties of plants with healthy and disease classes of egg-plant, hyacinth beans, ladies finger, and lime. The authors achieved the highest accuracy of 97.3% with GoogLeNet." Begum et al. [30] used three plant species of the PV dataset, peppers, potato, and tomato, for disease classification. The authors attained average accuracies of 94%, 95%, and 97% with the Xception, Inception Resnet-V2, and MobileNetV2 models. "The tomato plant fruit disease was categorized by [31] with VGG 16 and two ResNet models of

ResNet-50 and ResNet-101 into healthy and disease cases with a mean average precision of 90.87% from ResNet-101. Li et al. [32] achieved an accuracy of 95% with the same model with a different training based on CNN for the remotely sensed images."

In their work, Rangarajan et al. [17] classified tomato plant disease (TPD) with AlexNet and VGG16 for one healthy and six disease classes. Brahimi et al. [33] classified TPDs with AlexNet and the GoogLeNet model; for the PV dataset, they attained accuracies of 98.66% and 99.18%, respectively. Zhang et al. [34] used the ResNet-50 model to identify tomato leaf disease and achieved an accuracy of 97.28%. Karthik et al. [35], in their work for the detection of TPD, attained an accuracy of 95% with a residual CNN model and 98% with an attention-based residual CNN model. In detecting tomato plant leaves (TPL) with disease, Gonzalez et al. [36] used four models, MobileNetV2, NasNetMobile, Xception, and MobileNetV3, for the PV dataset and achieved accuracies of 75%, 84%, 100%, and 98%, respectively. Table 1 provides a comparative study of related work in plant disease classification.

Table 1. Comparative study of related work in classification of plant disease.

Ref No	Model	Objective	Dataset	Accuracy	Limitations
[22]	Ten-layer CNN	Classification of plant leaf	Flavia	87.92%	Flavia dataset consists of only healthy classes. Diseased classes are not studied.
[23]	AlexNet VGG19 VGG16 ResNet-50	Identification of plant disease	PlantVillage	83.66% 91.75% 94.96% 99.8%	Plant disease detection models can be deployed in mobile to help the farmers.
[26]	INC-VGGN	Classification of rice plant images	PlantVillage Own dataset	84.25% 91.83%	The size of the developed model is more to be used directly to be deployed on mobile as an App
[27]	ResNet-101	Classification of paddy leaf disease	Kaggle and UCI repository	91.52%	Other variety of paddy leaf diseases with a larger dataset and other CNN models can be used for better accuracy
[28]	Faster R-CNN	Diagnosis of rice plant disease	Kaggle and own dataset	98.25%	A mobile-based system with IoT can be implemented for future work.
[25]	AlexNet GoogLeNet VGG16 ResNet-101 DenseNet 201	Classification of soyabean plant disease	PlantVillage	95% 96.4% 96.4% 92.1% 93.6%	To develop a CNN model for better classification accuracy
[30]	Xception Inception Resnet-V2 MobileNetV2	Classification of plant disease	PlantVillage	94% 95% 97%	More classes can be used for the classification problem.

Table 1. *Cont.*

Ref No	Model	Objective	Dataset	Accuracy	Limitations
[29]	ResNet-101 GoogLeNet	Classification of ten different diseases in four crops	Own dataset	96.9% 97.3%	Dataset with a complex background can be used for classification.
[33]	AlexNet GoogLeNet	Classification of tomato plant disease	PlantVillage	98.66% 99.18%	The computation and size of the classification model can be reduced.
[17]	AlexNet VGG16	Classification of tomato plant disease	PlantVillage	97.49% 97.29%	The VGG16 model is computationally intensive.
[34]	ResNet-50	Identifying tomato leaf disease	PlantVillage	97.28%	The classification model can be used for detecting more variety of disease classes.
[35]	Attention-based Residual CNN	Detection of tomato leaf disease	PlantVillage	98%	More disease classes can be used in the future to detect disease.
[36]	MobileNetV2 NasNetMobile Xception MobileNetV3	Disease detection in tomato plant leaves	PlantVillage	75% 84% 100% 98%	Xception model is performing as the best classifier with high computation cost.

The PV database consists of 38 different plants leaves with healthy and disease classes of 14 species [37]. This paper offers the TPD classification with the proposed compact CNN models. In this work, a database with TPDs that occur in Indian states and the healthy type was chosen for analysis. Classification of nine leaf classes consisting of "Tomato Healthy (H) and disease classes Bacterial Spot (BS), Early Blight (EB), Late Blight (LB), Leaf Mold (LM), Mosaic Virus (MV), Septoria Leaf Spot (SLS), Target Spot (TS), and Yellow Leaf Curl Virus (YLCV)" was performed. The performances of the developed models are compared with that of ResNet-101 with transfer learning. The proposed models have less depth as compared to ResNet-101. The main contributions of this work are as follows:

1. Three highly accurate and compact models, N1, N2, and N3, have been proposed for the disease classification of TPL. The proposed models show high classification accuracy and require short training times. The performances of the models were validated by employing them to classify TPL from the challenging PV dataset and KVKN dataset. The models exhibited high classification accuracy for an unknown dataset.
2. The proposed models maintained good classification accuracy with compact model size. N1 and N3 were 8.5 MB in size, and N2 model was 17.14 MB.
3. To validate the versatility of the proposed models, they were also employed in tomato leaf disease classification using images captured from a mobile phone. The disease classification accuracy shows that the proposed models are well suited for TPL disease classification.

The paper describes the materials and methods in Section 2, followed by results and discussions in Section 3, and the conclusion in Section 4.

2. Materials and Methods

This research involved the classification of TPDs and the validation of the trained model with unknown data. Figure 1 depicts the workflow for classifying nine classes of TPL.

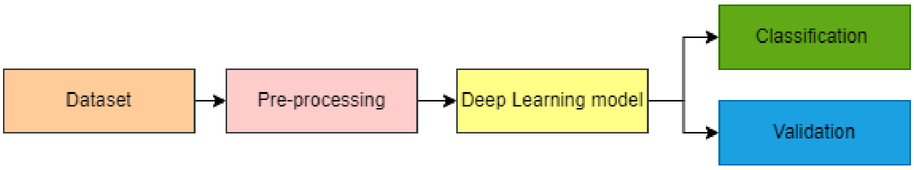


Figure 1. Workflow for classification and validation of TPDs.

2.1. Dataset and Pre-Processing

The TPL images from the PV database were used in this work [37]. The healthy tomato class and eight diseased leaf categories found in Indian states were used for classification purposes. Classification of nine leaf classes consisting of "Tomato Healthy (H) and disease classes Bacterial Spot (BS), Early Blight (EB), Late Blight (LB), Leaf Mold (LM), Mosaic Virus (MV), Septoria Leaf Spot (SLS), Target Spot (TS), and Yellow Leaf Curl Virus (YLCV)" was done. "It is critical to adhere to the basic steps that are customary in the study, one of which is pre-processing, for the actual maneuver of any algorithm and the preservation of uniformity in the study" [34,38–41]. The dataset was augmented with color augmentation of saturation, hue, and contrast; position augmentation of rotation by 45°, 135°, 225°, and 315°; and flipping horizontally and vertically, during the pre-processing stage. Saturation augmentation modifies the image's vibrancy. A grayscale image is fully desaturated, a partially desaturated image has muted colors, and positive saturation shifts colors closer to the primary colors. Adjusting the saturation of an image can help your model perform better. Hue augmentation changes the color channels of an input image at random, causing a model to consider alternative color schemes for objects and scenes in the input image. This technique is helpful in ensuring that a model does not memorize the colors of a given object or scene. Hue augmentation allows a model to consider the edges and shapes of things, and the colors. The degree of separation between an image's darkest and brightest areas is defined as contrast. The dataset is augmented with said combination of color and position augmentation. The augmented dataset consisted of 94,500 images resized to a standard size of 256 × 256 × 3 for the developed N1, N2, and N3 models and 224 × 224 × 3 for the ResNet-101 model. Table 2 shows the PV dataset images for each class before and after data augmentation. The KVKN dataset was used for predicting the performances of the trained models. The authors collected the data on the farm of KVKN, which were not augmented.

Table 2. Class-wise image data before and after augmentation of PV database.

Class	PV Database	
	Before Augmentation	**After Augmentation**
"BS"	100	10,500
"EB"	100	10,500
"H"	100	10,500
"LB"	100	10,500
"LM"	100	10,500
"MV"	100	10,500
"SLS"	100	10,500
"TS"	100	10,500
"YLCV"	100	10,500
TOTAL	900	94,500

2.2. CNN Models

The primary goal of this research was to develop a computationally less complex and precise "learning model" for classifying TPL. Figure 2 depicts the proposed compact CNN model for the classification and validation of the TPD. The three CNN models have variations in the Conv2D layer, as shown in Table 3. There are three sets of convolution 2D layers, "Conv2D layer," "batch normalization layer," and "ReLU layer." The "max-pooling layer follows the first two sets"; the "fully connected layer, softmax classifier, and classification layer" follow the third set.

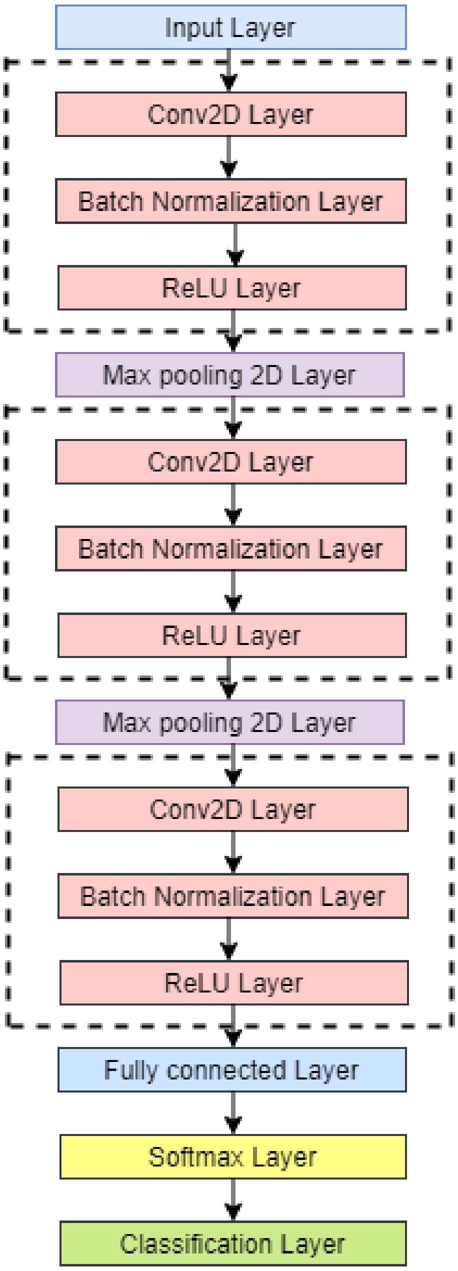


Figure 2. Proposed compact CNN model for classification.

The functional descriptions of convolutional layers for the developed CNN model 1 (N1), model 2 (N2) and, model 3 (N3) are as shown in Table 3.

Table 3. Functional description of convolution layers for N1, N2, and N3 models.

CNN Layer	CNN Model		
	N1	**N2**	**N3**
1st Conv2D	3 × 3, 8	3 × 3, 16	7 × 7, 8
2nd Conv2D	3 × 3, 16	3 × 3, 32	5 × 5, 16
3rd Conv2D	3 × 3, 32	3 × 3, 64	3 × 3, 32

"The convolutional layer describes a collection of the filters carrying out convolution over the entire image. Each convolutional layer learns the numerous features that detect discriminatory outlines in the tomato leaves to distinguish the type of disease in this architecture. CNN's feature extractor comprises particular neural networks that decide on weights during the training process. Deep neural networks see diverse feature evidence from the preceding layer after each gradient apprises a dataset. Furthermore, as the parameters of the initial layers are restructured through the training phase, the data delivery of this input feature map differs significantly. This significantly impacts training speed and necessitates various heuristics to determine parameter initialization [35]. This model employs an activation function known as the rectified linear unit (ReLU). It is the identity function, f(x) = x, for all positive values of input 'x,' and zeros for the negative values. ReLU is triggered sparingly, mimicking the neuron's inactivity in response to certain impulses. The neural network classification then operates on the image features and generates the output. The pooling layer activates only a subset of the feature map neurons. A '2-by-2' window is used across all blocks with a stride factor of '2.' The feature maps' width and height are effectively reduced while the number of channels remains constant. The neural network includes convolution layer piles and sets of pooling layers for feature extraction. The convolution layer employs the convolution process to transform the image. It is best described as a series of digital filters. The pooling layer combines neighboring pixels into a single pixel. The pooling layer then reduces the image dimensions. Batch normalization significantly reduces training time by normalizing the input of each layer in the network, not just the input layer. This method allows for higher learning rates, which reduces the number of training steps required for the network to converge [42]. The softmax function is the activation function in the CNN model's output layer that predicts a multinomial probability distribution."

"The benefits of small filter sizes over fully connected networks are that they minimize computing costs and weight sharing, resulting in lower back-propagation weights. Until now, the best choice for practitioners has been 3 × 3 [43,44]. The N1 CNN model has a fixed filter size of 3 × 3 in all three convolution layers. In the 1st Conv2D, there are eight filters, and in the 2nd Conv2D and 3rd Conv2D, there are 16 and 32 filters, respectively. In the N2 CNN model, the filter size is 3 × 3, and the number of filters in them is doubled compared to N1. In the N3 CNN model, the filter size for the 1st Conv2D layer is 7 × 7, with eight filters. The 2nd Conv2D layer is 5 × 5, with 16 filters, and the 3rd Conv2D layer is 3 × 3 with 32 filters."

The VGG16 model [45] is a 16-layer CNN model. The VGG16 model is shown in Figure 3. The VGG16 model has a convolutional layer, followed by a ReLU activation layer. All the convolutional layers have a filter size of 3 × 3 but a specific number of filters for the convolution. "The max-pooling layer follows two sets of two convolutional layers and one ReLU layer combinations. The max-pooling layer follows the next three sets of three convolutional layer and one ReLU layer combinations; these layers are followed by the fully connected layer, softmax layer, and classification layer." The proposed N1, N2, and N3 models have batch normalization layers.

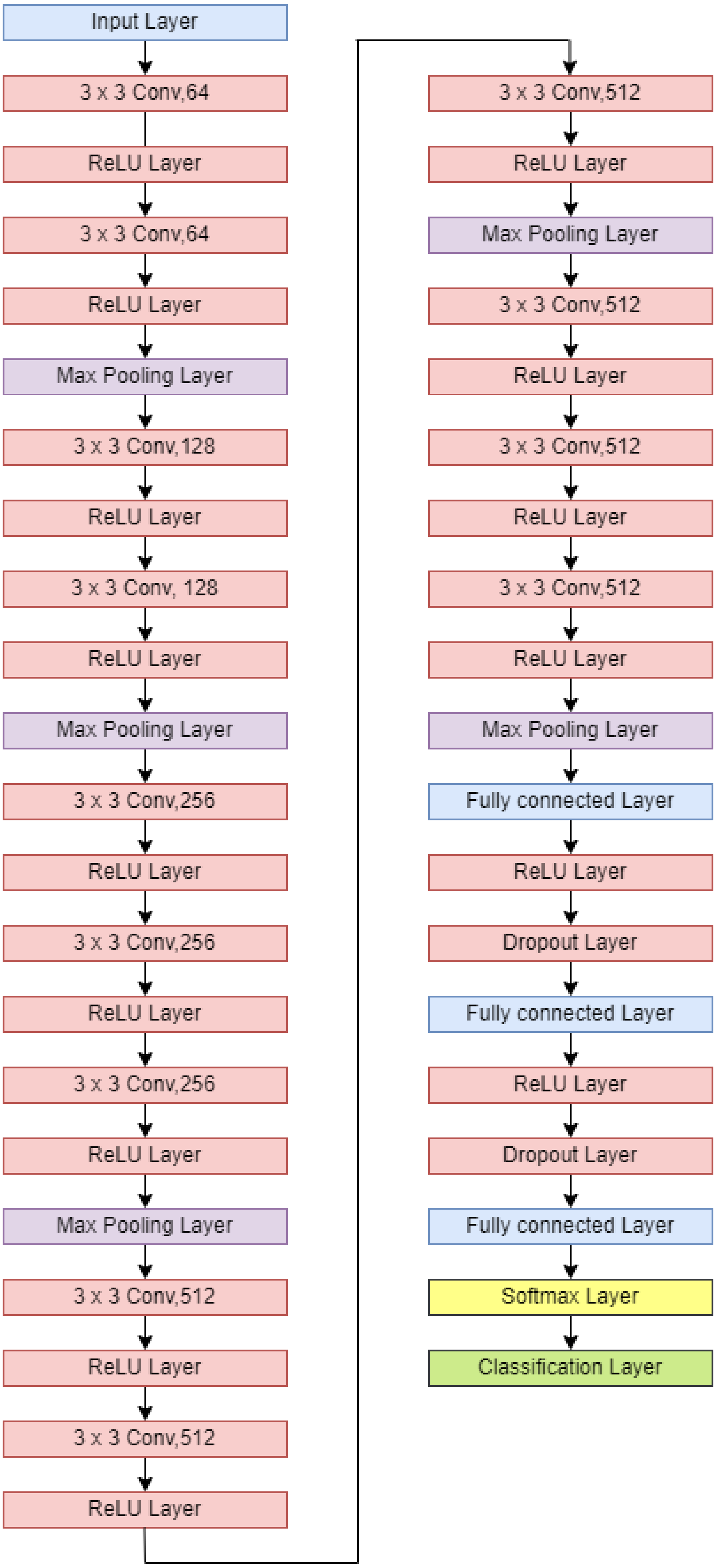


Figure 3. VGG16 model.

The top-5 error for the ResNet-50 model is 5.25; for ResNet-101 it is 4.60; and for ResNet-152 it is 4.49 [11]. ResNet-101 performs between ResNet-50 and ResNet-152, so ResNet-101 was chosen for the classification in this work. We used the ResNet-101 model with transfer learning and proposed N1, N2, and N3 models to classify nine TPL classes. This augmented dataset was used to train CNN models for TPL classification. This work's models were created in MATLAB2019b using a deep learning toolbox. The dataset was split into training and testing datasets, 80–20%, involving healthy and diseased plant leaves.

2.3. The CNN Model

"The classification of the model is based on the performance and its accuracy. The confusion matrix of the test dataset is used to evaluate the performance parameters." The diagonal elements of the confusion matrix show correct classification, and non-diagonal elements show misclassification. The following are the metrics [46,47]:

- "True positives (TP) represent the positive samples that were correctly labeled by the classifier,"
- "True negatives (TN) represent the negative samples correctly labeled by the classifier,"
- "False positives (FP) represent the negative samples incorrectly labeled as positive," and
- "False negatives (FN) correctly labeled the positive samples incorrectly labeled as negative."

"The performance parameters evaluated were macro-recall, macro-precision, macro-F1-score, and mean accuracy. Sensitivity/recall is the measure of the model that appropriately detects the positive class and is also known as the true positive rate. The model assigning positive events to the positive class is measured by a positive predictive value, also known as precision. F1-score is the harmonic mean of recall and precision. Macro-recall is the average per-class effectiveness of a classifier at identifying class labels. Macro-precision is the average agreement of data class labels per class with classifiers. Macro-F1-score is the relation between positive labels of the data and those agreed to by the classifier based on per-class average. Accuracy is the ratio of correct predictions to all predictions."

$$``Sensitivity/Recall = \frac{TP}{TP + FN}" \tag{1}$$

$$``MacroRecall = \frac{\sum_{n=1}^{C} Recall}{C}" \tag{2}$$

where C represents the number of classes.

$$``Precision = \frac{TP}{TP + FP}" \tag{3}$$

$$``MacroPrecision = \frac{\sum_{n=1}^{C} Precision}{C}" \tag{4}$$

$$``F1score = \frac{2XPrecisionXRecall}{PrecisionXRecall}" \tag{5}$$

$$``MacroF1score = \frac{\sum_{n=1}^{C} F1Score}{C}" \tag{6}$$

$$``Accuracy = \frac{TP + TN}{TP + TN + FP + FN}" \tag{7}$$

2.4. Validation of the Trained CNN Model

Following classification, the CNN models were validated using images from the PV database that were not included in the training or testing sets and images taken at KVKN.

Models were validated using 1090 images, which aided in predicting the class and accuracy of unknown data.

3. Results and Discussion

The entire investigation was carried out on an augmented dataset of 94,500 images from the PV database for nine Tomato plant classes. Figure 4 depicts healthy and diseased TPL images.

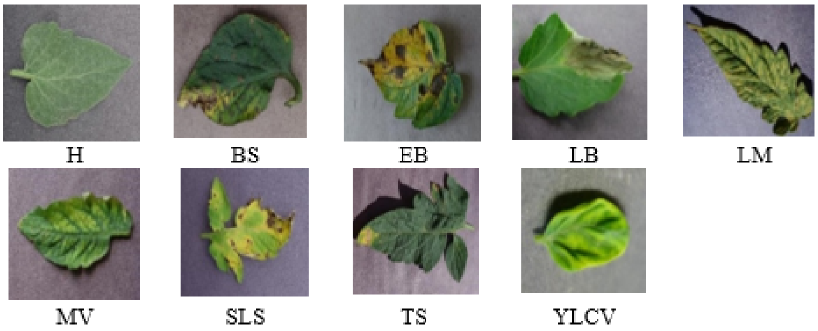


Figure 4. Images of TPL from the PV dataset.

As per the pre-processing in Section 2.1, data augmentation and image resizing were performed. Transformations were used to increase the data to avoid overfitting the training models and generalizing their responses. The dataset was augmented with color augmentation of saturation, hue, and contrast; position augmentation of rotation by 45°, 135°, 225°, and 315°; and flipping horizontally and vertically. Figure 5 shows some of the pre-processed images of TPL with color augmentation of hue, saturation, and contrast. The first row shows the original images of different TPL classes. The images in the second, third, and fourth row show the saturation, hue, and contrast augmentation, respectively, of the images in row one.

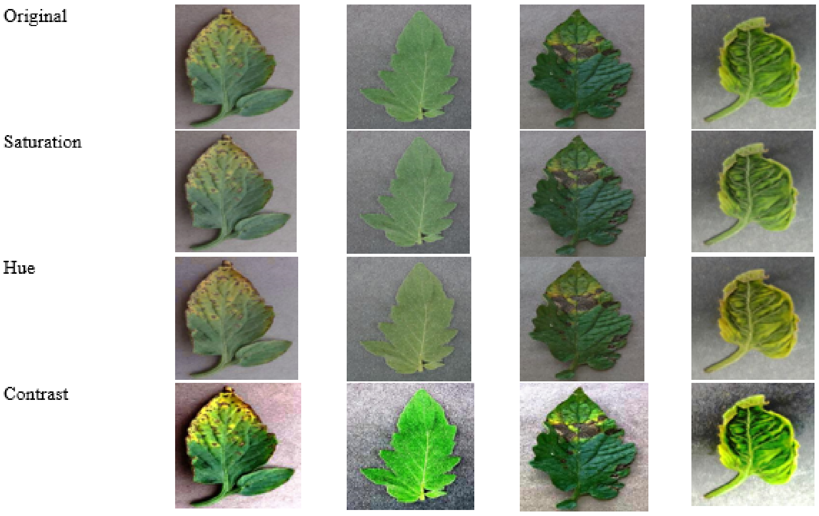
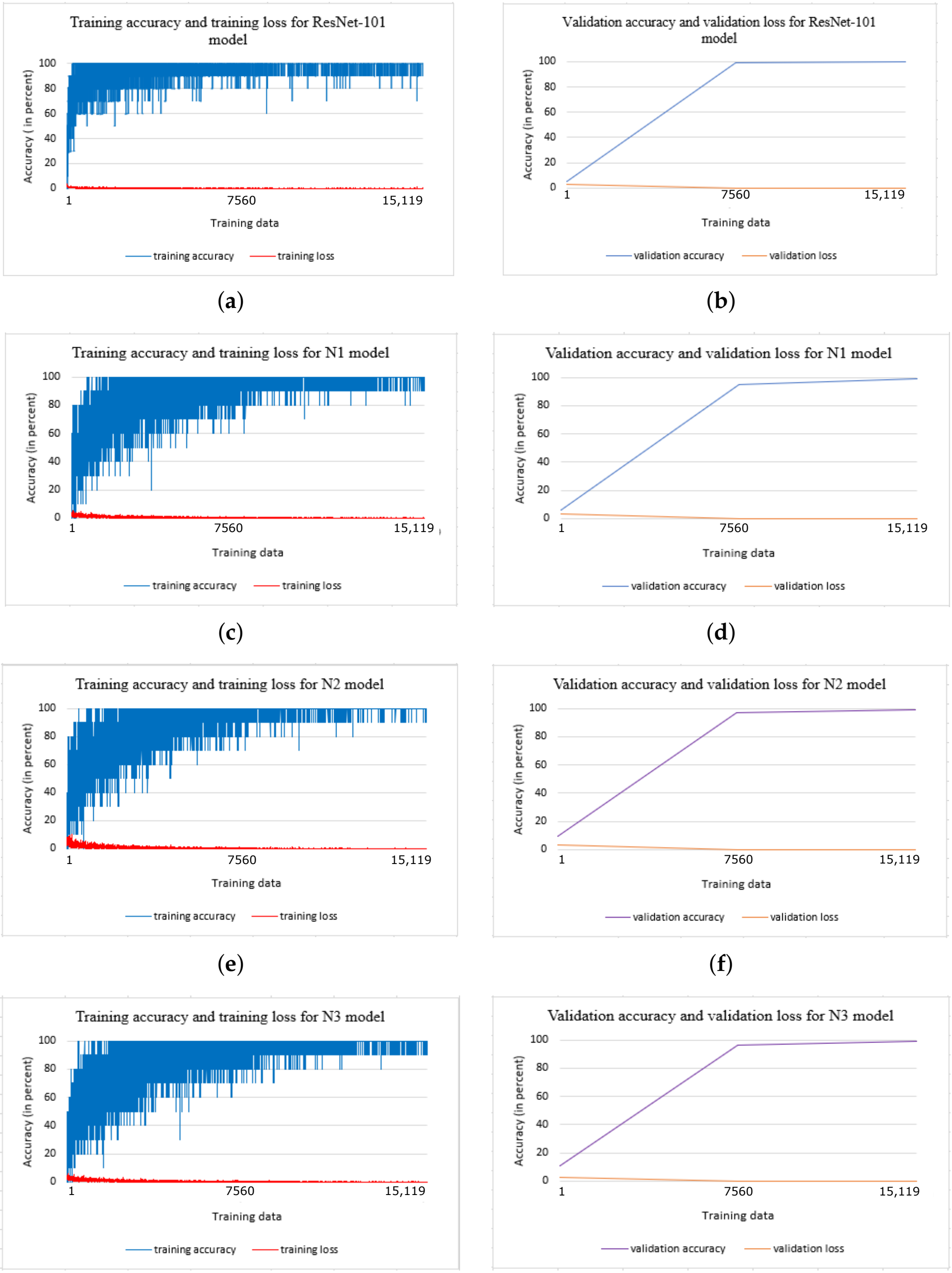


Figure 5. Pre-processed samples of a dataset of TPL.

"Overfitting occurs when the model fits well to training data but does not generalize well to new, previously unseen data. Overfitting problems can be prevented by taking measures such as data augmentation, simplifying the models, dropout, regularization, and early stopping" [48,49]. To ensure consistency, all networks used here had the same hyperparameters. In this work, the mini-batch size was set to 10, the epochs were set to 2, and the learning rate was set to 0.0001. "The training loss (TL) was reduced to a minimum value in two epochs. Hence, two epochs was chosen. The training accuracy (TA) and TL, along with the validation accuracy (VA) and validation loss (VL), were as shown in Figure 6 for the ResNet-101, N1, N2, and N3 models. The model with increasing TA and VA and decreasing TL and VL show that overfitting was prevented. The TA and TL for the models are shown in Figure 6a for ResNet-101, Figure 6c for N1, Figure 6e for N2, and Figure 6g for N3. The VA and VL for the models are shown in Figure 6b for the ResNet-101 model, Figure 6d for the N1 model, Figure 6f for the N2 model, and Figure 6h for the N3 model."



(**a**) (**b**)

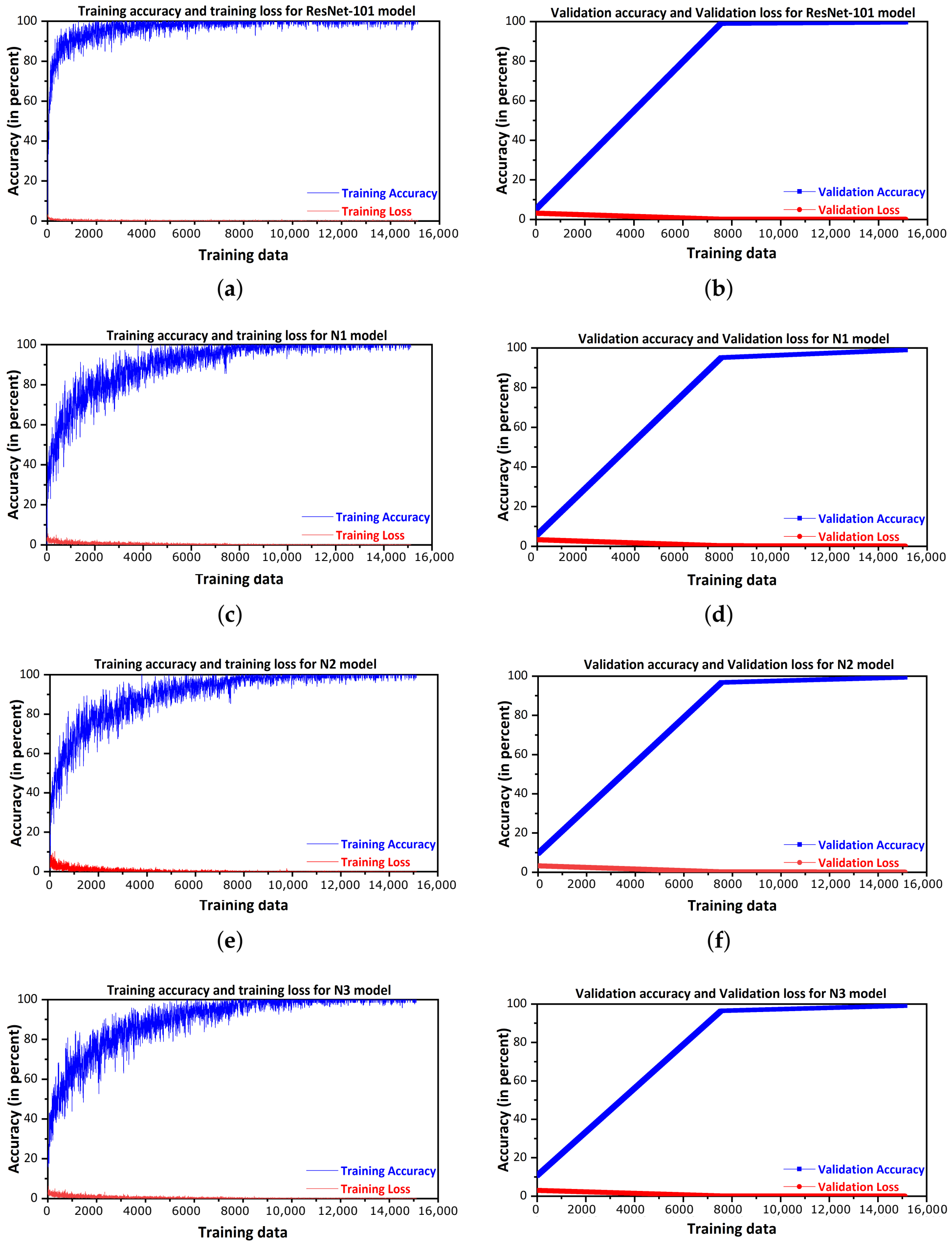
(**c**) (**d**)

(**e**) (**f**)

(**g**) (**h**)

Figure 6. TA and TL and VA and VL were calculated for the ResNet-101 and N1, N2, and N3 models.

"Smoothing the graph allows important patterns to stand out more clearly. The smoothened graphs of TA and TL, and VA and VL for the ResNet-101 and proposed N1, N2, and N3 models are shown in Figure 7. The TL curves for the models are shown in Figure 7a for ResNet-101, Figure 7c for N1, Figure 7e for N2, and Figure 7g for N3. The VA and VL for the models are shown in Figure 7b for ResNet-101, Figure 7d for N1, Figure 7f for N2, and Figure 7h for N3."



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

(**g**) (**h**)

Figure 7. TL, VA, and VL were calculated for the ResNet-101 and N1, N2, and N3 models.

The tomato plant images were classified with the ResNet-101 model with transfer learning and the proposed N1, N2, and N3 models, as shown in Figure 8. For TPD classification, each model was trained with 80% of the dataset and tested with 20% of the dataset. The classified images with the N1 model are shown in Figure 8a. Figure 8b depicts the N2-classified images. Figure 8c illustrates images classified by the N3 model, and Figure 8d shows images classified by the ResNet-101 model.

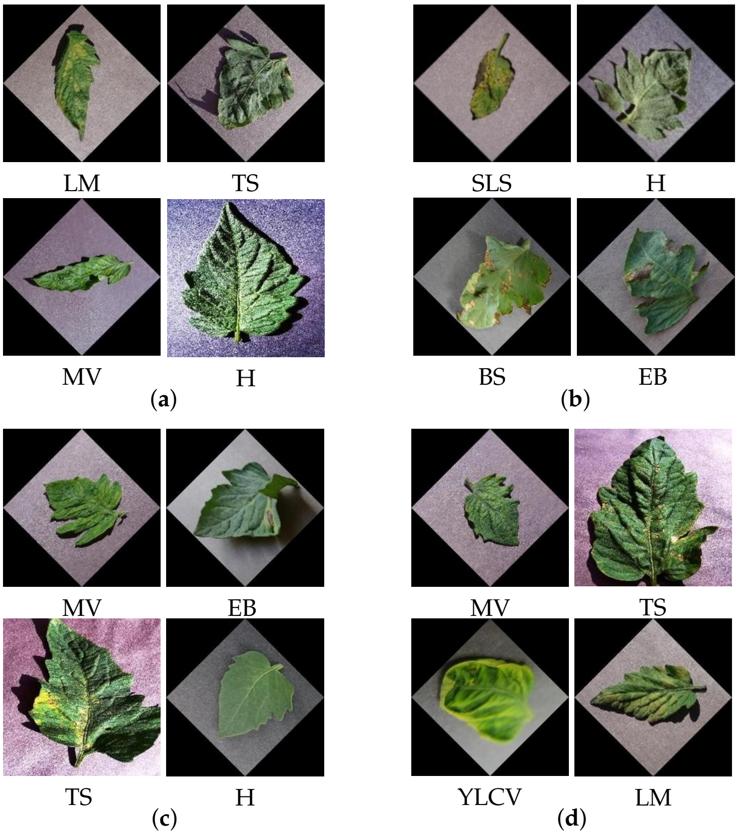


Figure 8. Output images after classification for 20% testing data using (**a**) N1, (**b**) N2, (**c**) N3, (**d**) ResNet-101.

Table 4 shows the classification accuracies of N1, N2, N3, and ResNet-101 models for 80% of the training dataset. The table compares previous work on plant leaves' classification with the proposed work. Brahimi et al. [33] attained accuracies of 98.66% and 99.18% for AlexNet and GoogLeNet, respectively. N1, N2, and N3 achieved accuracies of 99.13%, 99.51%, and 99.40%, respectively. The input size of images for AlexNet was 227 × 227 × 3, and it was 224 × 224 × 3 for GoogLeNet, VGG 16, and ResNet models. The ten-layer CNN model was fed with images of size 64 × 64 × 3, and the attention-based residual CNN was 256 × 256 × 3. All the models compared in Table 4 were trained with the PV database. "The ten-layer CNN model by [22] achieved accuracy of 87.92% with the Flavia dataset and 84.02% with the PV dataset." When identifying TPL disease, Anandhakrishnan et al. [50] achieved an accuracy of 99.45% with the Xception V4 model. Qiu et al. [8], in their work on plant disease recognition on a self-collected dataset, achieved an average accuracy of 97.62%. The VGG16 model was used to train a "teacher model" with a better recognition rate and a much larger volume than the "student model." The information was then transferred to MobileNet via distillation. This process reduced the model size to 19.83 MB. The classification accuracy for the pretrained VGG16 model was 99.21%, and the size of the trained model was 477 MB. The proposed trained models N1 and N3 were 8.5 MB, and the N2 model was 17.14 MB. The pre-trained ResNet-101 demonstrated classification accuracy of 99.97%, and the size of the trained model was 151 MB. AlexNet, GoogLeNet, and VGG 16 had larger model sizes as compared to the N1, N2, and N3 models. The developed N2 model achieved accuracy in the same range as ResNet-101 and VGG16, though being 88.65% smaller than ResNet-101 and 96.41% smaller than VGG16.

Table 4. Performance comparison of proposed work in comparison to other existing works.

Model & Ref No.	Datasize	Accuracy	Model Size
AlexNet [33]	14,828	98.66%	227 MB [51]
GoogLeNet [33]	14,828	99.18%	27 MB [51]
AlexNet [17]	13,262	97.49%	227 MB [51]
VGG16 [17]	13,262	97.29%	515 MB [51]
ResNet [34]	41,127	97.28%	96 MB [51]
Ten-layer CNN [22]	94,500	84.02%	7 MB
Attention based Residual CNN [35]	95,999	98%	Not given
Xception V4 [50]	14,528	99.45%	85 MB [51]
Distilled MobileNet [52]	54,305	97.62%	19.83 MB
VGG16	94,500	99.21%	477 MB
N1	94,500	99.13%	8.5 MB
N2	94,500	99.51%	17.14 MB
N3	94,500	99.40%	8.5 MB
ResNet-101	94,500	99.97%	151 MB

The CNN model's training time is also critical, as illustrated in Figure 9. The N1, N2, and N3 models are three-layer CNN models that are compact in size. The N1 model takes less time than the N2 model. The N2 model has twice the number of filters as the N1 model. The VGG16 model also took more training time than the proposed models. There was a steep rise in the training time for the ResNet-101 model. Note that 89% of the time was reduced for training the N2 model compared to the ResNet-101 model. The proposed models have shown better results than state-of-the-art classifiers.

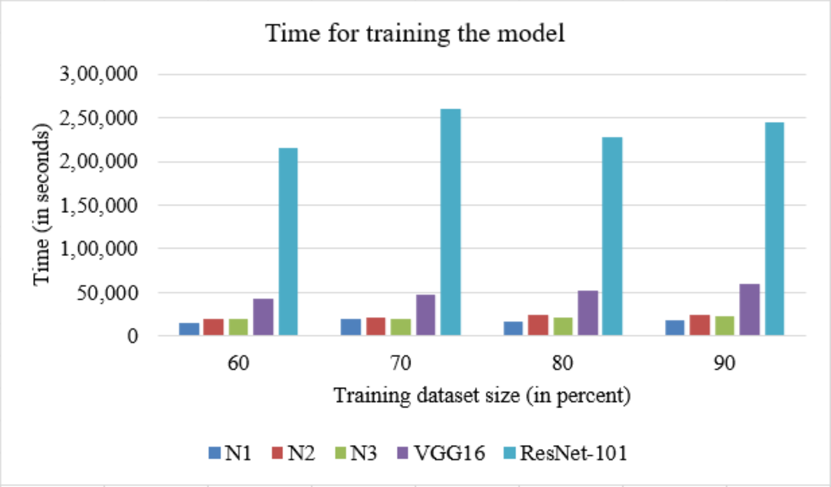


Figure 9. Training time for deep learning networks with varying training dataset sizes.

The confusion matrix contains information on the correct and incorrect classification of each of the nine classes of tomato leaves. Table 5 shows the confusion matrix for the ResNet-101 and proposed N1, N2, and N3 models.

Table 5. Confusion matrix for proposed models for the PV dataset.

(a) "Confusion matrix for ResNet-101 model."

		Actual Class								
Predicted Class	Class	BS	EB	H	LB	LM	MV	SLS	TS	YLCV
	BS	2097	0	0	0	0	0	1	0	2
	EB	0	2100	0	0	0	0	0	0	0
	H	0	0	2100	0	0	0	0	0	0
	LB	0	0	0	2099	0	0	1	0	0
	LB	0	0	0	0	2100	0	0	0	0
	MV	0	0	0	0	0	2100	0	0	0
	SLS	0	0	0	0	0	0	2100	0	0
	TS	0	0	2	0	0	0	0	2098	0
	YLCV	0	0	0	0	0	0	0	0	2100

(b) Confusion matrix for N1 model

		Actual Class								
Predicted Class	Class	BS	EB	H	LB	LM	MV	SLS	TS	YLCV
	BS	2077	4	0	3	0	0	13	3	0
	EB	0	2064	1	6	5	0	19	3	2
	H	0	0	2100	0	0	0	0	0	0
	LB	0	0	0	2094	0	1	2	0	3
	LB	2	2	0	2	2078	2	6	7	1
	MV	0	0	3	3	0	2090	4	0	0
	SLS	4	0	0	11	2	0	2083	0	0
	TS	0	0	1	0	0	0	4	2095	0
	YLCV	4	5	0	9	2	3	16	7	2054

(c) Confusion matrix for N2 model

		Actual Class								
Predicted Class	Class	BS	EB	H	LB	LM	MV	SLS	TS	YLCV
	BS	2080	4	0	1	3	0	6	0	6
	EB	0	2097	0	1	0	0	1	1	0
	H	0	0	2097	2	0	0	1	0	0
	LB	0	0	1	2076	6	2	6	1	8
	LB	1	2	0	2	2074	0	11	0	10
	MV	0	0	0	0	0	2095	4	0	1
	SLS	2	0	0	1	0	2	2092	0	3
	TS	0	0	1	0	0	0	1	2098	0
	YLCV	0	0	0	1	1	0	0	0	2098

(d) Confusion matrix for N3 model

		Actual Class								
Predicted Class	Class	BS	EB	H	LB	LM	MV	SLS	TS	YLCV
	BS	2074	3	0	3	4	1	8	2	5
	EB	2	2090	0	3	1	1	1	1	1
	H	0	0	2099	1	0	0	0	0	0
	LB	0	2	2	2090	0	1	3	0	2
	LB	2	2	0	7	2080	2	7	0	0
	MV	0	0	0	0	0	2096	4	0	0
	SLS	0	1	1	5	4	2	2083	1	3
	TS	0	0	3	0	0	0	0	2097	0
	YLCV	1	2	2	5	2	6	3	1	2078

The performance parameters were calculated based on the elements of the confusion matrix. Based on this, the performance parameters for N1, N2, N3, and ResNet-101 models are shown in Table 6. Brahimi et al. [33] stated that "the mean accuracy was 99.18% for GoogLeNet." The average precisions by [31] for classification of disease in tomato fruit using VGG16, ResNet-50, and ResNet-101 were 88.28%, 89.53%, and 90.87%, respectively. The proposed N1, N2, N3, and ResNet-101 models achieved macro-precision of 99.13%, 99.51%, 99.40%, and 98.10%, respectively. The proposed N1, N2, and N3 models achieved mean accuracies of 99.81%, 99.89%, 99.86%, respectively, and a mean accuracy of 99.58% for the ResNet-101 model. Macro-recall, macro-precision, and macro-F1-score of N1, N2 and N3 are higher than those of the ResNet-101 model.

Table 6. Performance parameters for the classification of nine classes of Tomato plant.

"Model"	**"Macro Recall"**	**"Macro Precision"**	**"Macro F1 Score"**	**"Mean Accuracy"**
N1	99.13%	99.13%	99.13%	99.81%
N2	99.51%	99.51%	99.51%	99.89%
N3	99.4%	99.4%	99.4%	99.86%
ResNet-101	98.11%	98.1%	98.09%	99.58%

The performance parameters of recall, precision, F1-score, and accuracy for all nine classes of TPL for the proposed N1, N2, N3 model, and ResNet-101 model are shown in Figures 10–13. The recall for the nine classes is shown in Figure 10. The performance of the N2 model is suitable for all classes. The recall values for EB, SLS, and TS classes are low for the ResNet-101 model compared to the proposed N1, N2, and N3 models. Precision for all the nine classes of TPL was evaluated and is shown in Figure 11. The F1-score is shown in Figure 12 for all nine classes. The ResNet-101 model showed lower performances for EB, LB, LM, SLS, and TS. Accuracy for the TPL classes is shown in Figure 13. The accuracy for all classes was good for the proposed N1, N2, and N3 models.

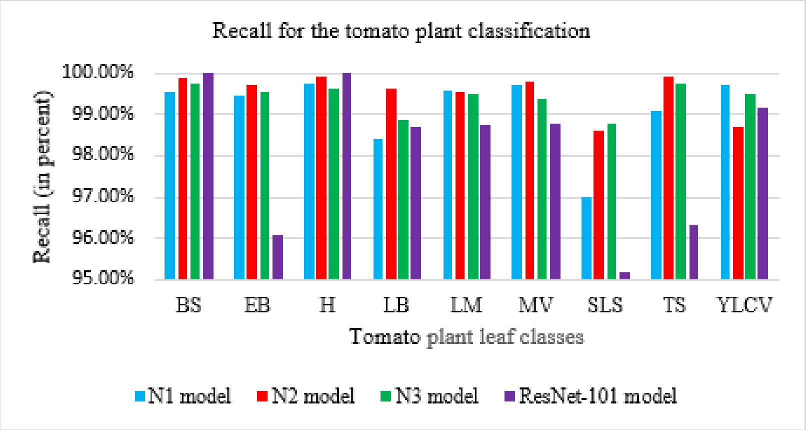


Figure 10. Recall performance for TPL classes.

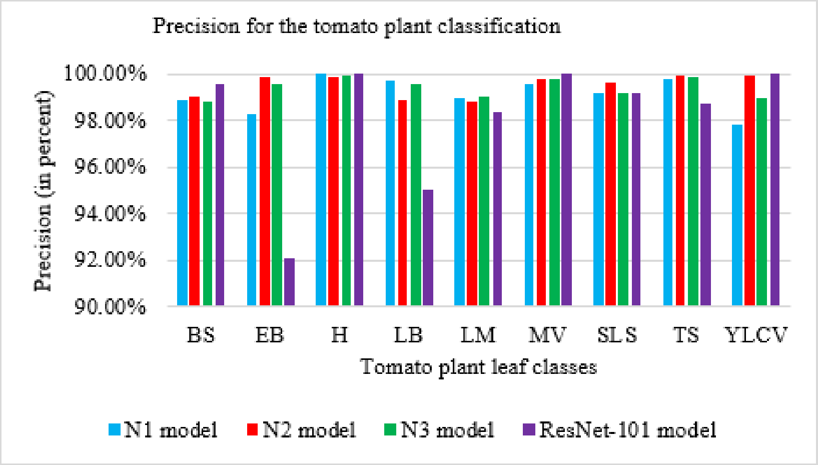


Figure 11. Precision performance for TPL classes.

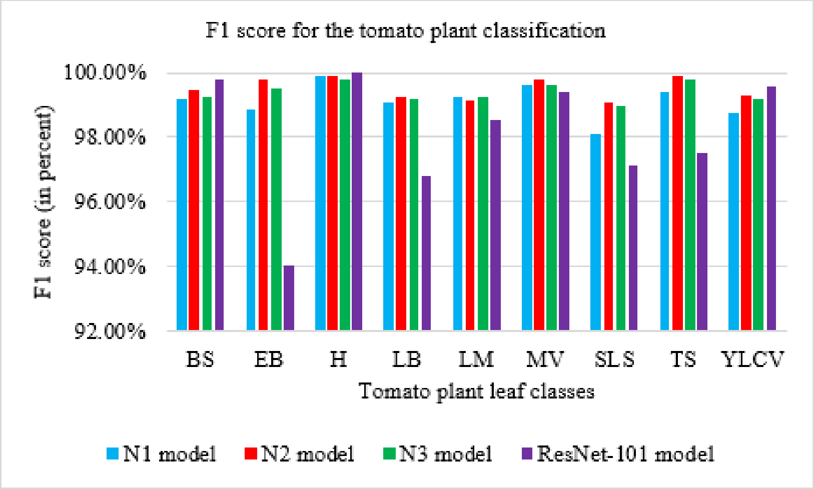


Figure 12. F1-score performance for TPL classes.

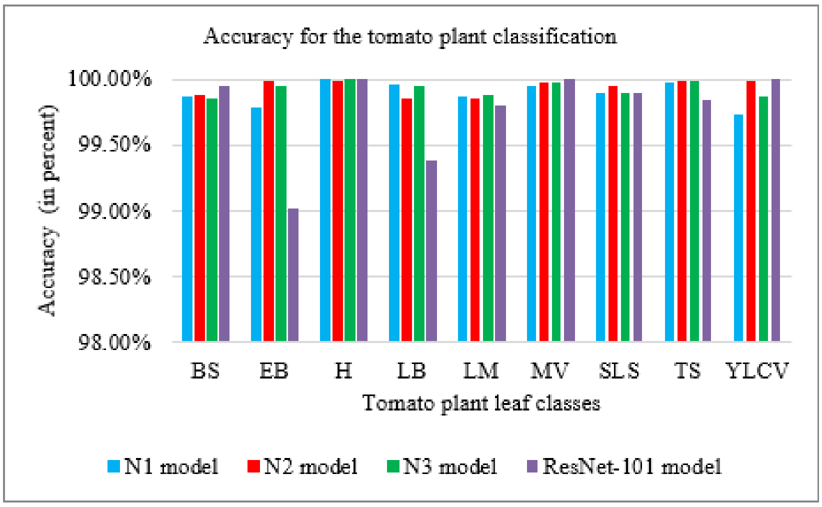


Figure 13. Accuracy performance for TPL classes.

As per the pre-processing section, The receiver operating characteristic (ROC) explicitly states how well the probabilities of the positive and negative classes are distinguished. The ROC curve is generated by varying this probability threshold and computing the corresponding true positive rate (TPR) and false positive rate (FPR). The x-axis in ROC represents the FPR, and the y-axis represents the TPR [53,54]. The area under the curve (AUC) is a critical calculation metric for assessing the performance of any classification model. It indicates how well the model can distinguish between classes. The higher the AUC, the more accurately the model predicts the classes. ROC is a probability curve, and the AUC represents the degree or measure of separability. "The AUC is a measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the model is at distinguishing between the positive and negative classes. The ROC curves for the ResNet-101 and proposed N1, N2, and N3 models are shown in Figure 14." The AUC for ResNet-101 and N2 is 100%, and the AUC for N1 and N3 is 99.98%. This result shows the excellent performance of the N1, N2, N3, and ResNet-101 models in the classification of TPL.

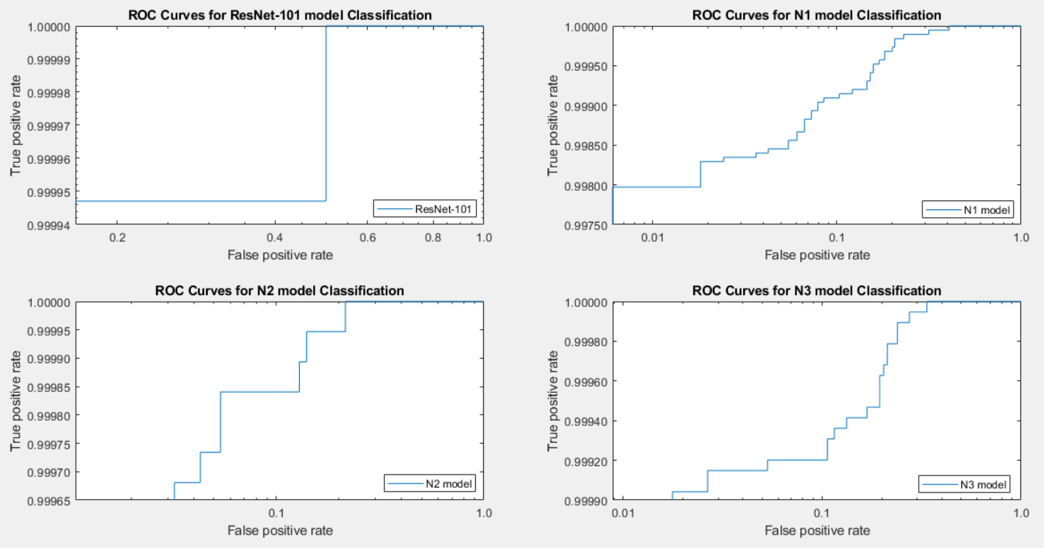


Figure 14. ROC for the ResNet-101, N1, N2, and N3 models.

The average precision (AP) is an essential parameter in a detection or classification task; it is the area under the precision–recall curve. The AP for the nine classes is shown in Table 7 for the PV dataset.

Table 7. Average precision of each class of tomato leaf classification.

Class	N1	N2	N3	ResNet-101
BS	98.9%	99.05%	98.76%	99.58%
EB	98.29%	99.86%	99.52%	92.08%
H	100%	99.86%	99.95%	100%
LB	99.71%	98.86%	99.52%	95%
LM	98.95%	98.76%	99.05%	98.33%
MV	99.52%	99.76%	99.81%	100%
SLS	99.19%	99.62%	99.19%	99.17%
TS	99.76%	99.9%	99.86%	98.75%
YLCV	97.81%	99.9%	98.95%	100%

The trained N1, N2, N3, and ResNet-101 models were validated with the anonymous data to predict its class and accuracy. The mean accuracy for validating the model is shown in Figure 15 for anonymous PV data. The N2 model delivered excellent performance for all the classes compared with N1 and N3 models. N2 showed mean accuracy of 96.40% for H and LB classes. EB had low accuracy for all the models. Overall, it can be seen that the N2 model behaved exceptionally well in classification and prediction, along with ResNet-101. Figure 16 shows the trained models' predictions of the KVKN captured image. The image class was predicted as LB by all the models, and the accuracy of prediction by N1 was 100%, that of N2 was 98.44%, that of N3 model was 96%, and that of ResNet-101 was 95.95%.

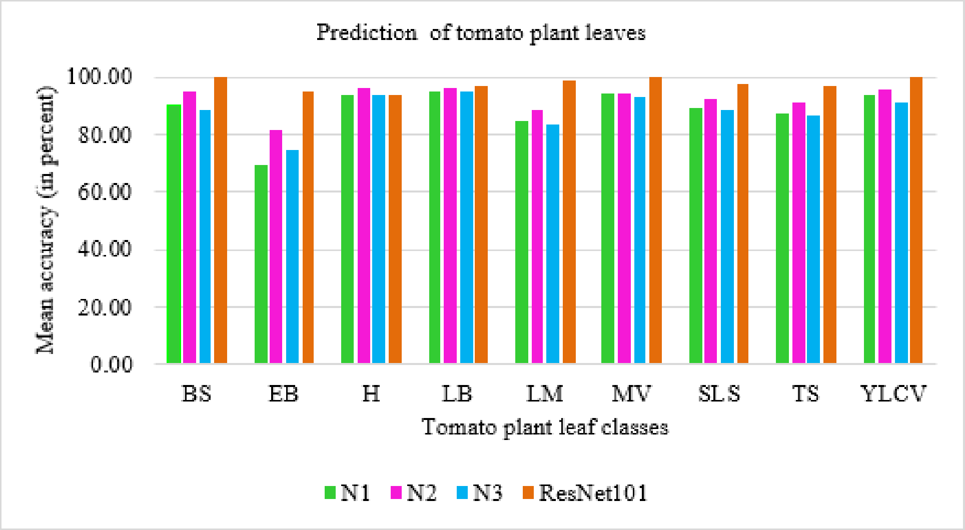
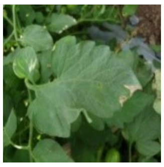


Figure 15. Predictions by N1, N2, N3, and ResNet-101 for PV data.



N1: 100% LB
N2: 98.44% LB
N3: 96% LB
ResNet-101: 95.95% LB

Figure 16. Predictions by N1, N2, N3, and ResNet-101 for KVKN data.

Computational models with robustness and high precision computing output have extended their usage in practical application scenarios, including classification in healthcare, industry, etc. The developed N1 model, N2 model, and N3 model trained on the PV dataset were able to predict the class of TPL of the KVKN dataset. The models can be deployed via applications on mobile phones in the future, allowing farmers to make quick decisions about tomato plant disease management. The management step towards infected plants can be spraying appropriate pesticides or just removing the infected plants from the field to avoid the further spread of disease.

4. Conclusions

This work used deep learning models, N1, N2, N3, and ResNet-101, to classify TPL images from the PV database. The developed model showed an accuracy of classification equally as good as ResNet-101. Compared to the ResNet-101 model, the developed model's training time was reduced by 92% for the N1 model, 89% for the N2 model, and 90% for the N3 model. N2 is 88.65% more compact than ResNet-101 and is about as accurate. The developed models outperform ResNet-101 in terms of performance parameters such as "macro-precision, macro-F1-score, and mean accuracy." The proposed N2 model had an AUC of 100%, and the N1 and N3 models have an AUC of 99.98%, indicating good classifier performance. The average precision in each tomato plant class has been consistently strong, affirming a more robust classification process. The PV and KVKN images were used to validate these trained models. The mean accuracies of N1, N2, N3, and ResNet-101 were 99.81%, 99.89%, 99.86%, and 99.58%, respectively, for the PV test dataset. The prediction accuracies for proposed N1, N2, N3, and ResNet-101 models were 100% LB, 98.44% LB, 96% LB, and 95.95% LB for the KVKN dataset. This classification problem will assist farmers in detecting and taking appropriate steps for disease management, which will benefit society.

In the future, the models can be deployed via an application to mobile phones that can help farmers make rapid decisions about the management of tomato plant diseases.

Author Contributions: Conceptualization, S.A.W. and H.R.; methodology, S.A.W.; software, S.A.W.; validation, H.R. and V.V.; formal analysis, S.A.W.; investigation, S.A.W.; resources, H.R. and V.V.; data curation, S.A.W.; writing—original draft preparation, S.A.W.; writing—review and editing, H.R., V.V. and K.K.; visualization, S.A.W.; supervision, H.R.; project administration, H.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank the officials at KVKN for allowing us to capture images of the tomato plants in the field.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. AlZu'bi, S.; Hawashin, B.; Mujahed, M.; Jararweh, Y.; Gupta, B.B. An efficient employment of internet of multimedia things in smart and future agriculture. *Multimed. Tools Appl.* **2019**, *78*, 29581–29605. [CrossRef]
2. Arivazhagan, S.; Shebiah, R.N.; Ananthi, S.; Varthini, S.V. Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. *Agric. Eng. Int. CIGR J.* **2013**, *15*, 211–217.
3. Al Bashish, D.; Braik, M.; Bani-ahmad, S. A Framework for Detection and Classification of Plant Leaf and Stem Diseases. In Proceedings of the International Conference on Signal and Image Processing, Chennai, India, 15–17 December 2010; pp. 113–118.
4. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [CrossRef]
5. Lee, S.H.; Chan, C.S.; Mayo, S.J.; Remagnino, P. How deep learning extracts and learns leaf features for plant classification. *Pattern Recognit.* **2017**, *71*, 1–13. [CrossRef]
6. Kumar, N.; Belhumeur, P.N.; Biswas, A.; Jacobs, D.W.; Kress, W.J.; Lopez, I.C.; Soares, J.V. Leafsnap: A Computer Vision System for Automatic Plant Species Identification. In Proceedings of the European Conference on Computer Vision, Firenze, Italy, 7–13 October 2012; pp. 502–516. [CrossRef]
7. Yann, L.; Leon, B.; Yoshua, B.; Patrick, H. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, 86, 2278–2324. [CrossRef]
8. Aquil, M.A.I.; Ishak, W.H.W. Evaluation of scratch and pre-trained convolutional neural networks for the classification of tomato plant diseases. *IAES Int. J. Artif. Intell.* **2021**, *10*, 467–475. [CrossRef]
9. Schmidt, K.W.; Hazir, O. A Data Envelopment Analysis Method for Finding Robust and Cost-Efficient Schedules in Multimode Projects. *IEEE Trans. Eng. Manag.* **2019**, *67*, 414–429. [CrossRef]
10. Elmousalami, H.H. Comparison of artificial intelligence techniques for project conceptual cost prediction. *IEEE Trans. Eng. Manag.* **2020**, *68*, 183–196. [CrossRef]
11. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
12. Raghu, S.; Sriraam, N.; Temel, Y.; Rao, S.V.; Kubben, P.L. EEG based multi-class seizure type classification using convolutional neural network and transfer learning. *Neural Netw.* **2020**, *124*, 202–212. [CrossRef]
13. Syarief, M.; Setiawan, W. Convolutional neural network for maize leaf disease image classification. *Telkomnika Telecommun. Comput. Electron. Control* **2020**, *18*, 1376–1381. [CrossRef]
14. Hu, X.; Xu, J.; Wu, J. A Novel Electronic Component Classification Algorithm Based on Hierarchical Convolution Neural Network. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Changsha, China, 18–20 September 2020; Volume 474, pp. 1–7. [CrossRef]
15. Al-Qerem, A.; Alauthman, M.; Almomani, A.; Gupta, B.B. IoT transaction processing through cooperative concurrency control on fog–cloud computing environment. *Soft Comput.* **2020**, *24*, 5695–5711. [CrossRef]
16. Abid, A.; Sinha, P.; Harpale, A.; Gichoya, J.; Purkayastha, S. Distributed Computing and Artificial Intelligence—9th International Conference. In *Optimizing Medical Image Classification Models for Edge Devices*; AISC: Springer, Berlin, Germany, 2021; Volume 151, pp. 77–87.
17. Rangarajan, A.K.; Purushothaman, R. Tomato crop disease classification using pre-trained deep learning algorithm. In Proceedings of the International Conference on Robotics and Smart Manufacturing, Chennai, India, 19–21 July 2018; Volume 133, pp. 1040–1047. [CrossRef]
18. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors* **2017**, *17*, 2022. [CrossRef]
19. Fuentes, A.F.; Yoon, S.; Lee, J.; Park, D.S. High-Performance Deep Neural Network-Based Tomato Plant Diseases and Pests Diagnosis System With Refinement Filter Bank. *Front. Plant Sci.* **2018**, *9*, 1162. [CrossRef]

20. Duan, C.Y.; Chen, X.Q.; Shi, H.; Liu, H.C. A New Model for Failure Mode and Effects Analysis Based on k-Means Clustering Within Hesitant Linguistic Environment. *IEEE Trans. Eng. Manag.* **2019**, *69*, 1837–1847. [CrossRef]
21. 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* **2020**, *9*, 1302. [CrossRef]
22. Liu, J.; Yang, S.; Cheng, Y.; Song, Z. Plant Leaf Classification Based on Deep Learning. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2019; pp. 3165–3169. [CrossRef]
23. Mukti, I.Z.; Biswas, D. Transfer Learning Based Plant Diseases Detection Using ResNet50. In Proceedings of the 2019 4th International Conference on Electrical Information and Communication Technology, EICT 2019, Khulna, Bangladesh, 20–22 December 2019; pp. 1–6. [CrossRef]
24. Wagle, S.A.; Harikrishnan, R. Comparison of Plant Leaf Classification Using Modified AlexNet and Support Vector Machine. *Traitement Signal* **2021**, *38*, 79–87. [CrossRef]
25. Jadhav, S.B.; Udupi, V.R.; Patil, S.B. Convolutional neural networks for leaf image-based plant disease classification. *IAES Int. J. Artif. Intell.* **2019**, *8*, 328–341. [CrossRef]
26. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanehkaran, Y.A. Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* **2020**, *173*, 105393. [CrossRef]
27. Islam, M.A.; Shuvo, N.R.; Shamsojjaman, M.; Hasan, S.; Hossain, S.; Khatun, T. An Automated Convolutional Neural Network Based Approach for Paddy Leaf Disease Detection. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 280–288. [CrossRef]
28. Bari, B.S.; Islam, M.N.; Rashid, M.; Hasan, M.J.; Razman, M.A.M.; Musa, R.M.; Ab Nasir, A.F.; Majeed, A.P.A. A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework. *PeerJ Comput. Sci.* **2021**, *7*, e432. [CrossRef]
29. Rangarajan Aravind, K.; Raja, P. Automated disease classification in (Selected) agricultural crops using transfer learning. *Automatika* **2020**, *61*, 260–272. [CrossRef]
30. Begum, A.S.; Savitha, S.; Shahila, S.; Sharmila, S. Diagnosis of Leaf Disease Using Enhanced Convolutional Neural Network. *Int. J. Innov. Res. Appl. Sci. Eng.* **2020**, *3*, 579–586. [CrossRef]
31. Wang, Q.; Qi, F. Tomato diseases recognition based on faster RCNN. In Proceedings of the 10th International Conference on Information Technology in Medicine and Education, ITME 2019, Qingdao, China, 23–25 August 2019; pp. 772–776. [CrossRef]
32. Li, W.; Liu, H.; Wang, Y.; Li, Z.; Jia, Y.; Gui, G. Deep Learning-Based Classification Methods for Remote Sensing Images in Urban Built-Up Areas. *IEEE Access* **2019**, *7*, 36274–36284. [CrossRef]
33. Brahimi, M.; Boukhalfa, K.; Moussaoui, A. Deep Learning for Tomato Diseases: Classification and Symptoms Visualization. *Appl. Artif. Intell.* **2017**, *31*, 299–315. [CrossRef]
34. Zhang, K.; Wu, Q.; Liu, A.; Meng, X. Can deep learning identify tomato leaf disease? *Adv. Multimed.* **2018**, *2018*, 6710865. [CrossRef]
35. Karthik, R.; Hariharan, M.; Anand, S.; Mathikshara, P.; Johnson, A.; Menaka, R. Attention embedded residual CNN for disease detection in tomato leaves. *Appl. Soft Comput. J.* **2020**, *86*, 105933. [CrossRef]
36. Gonzalez-Huitron, V.; León-Borges, J.A.; Rodriguez-Mata, A.E.; Amabilis-Sosa, L.E.; Ramírez-Pereda, B.; Rodriguez, H. Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4. *Comput. Electron. Agric.* **2021**, *181*, 105951. [CrossRef]
37. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using Deep Learning for Image-Based Plant Disease Detection. *Front. Plant Sci.* **2016**, *7*, 1419. [CrossRef]
38. Dyrmann, M.; Karstoft, H.; Midtiby, H.S. Plant species classification using deep convolutional neural network. *Biosyst. Eng.* **2016**, *151*, 72–80. [CrossRef]
39. Zhang, Y.; Phillips, P.; Wang, S.; Ji, G.; Yang, J.; Wu, J. Fruit classification by biogeography-based optimization and feedforward neural network. *Expert Syst.* **2016**, *33*, 239–253. [CrossRef]
40. Zhang, Y.D.; Satapathy, S.C.; Wang, S.H. Fruit category classification by fractional Fourier entropy with rotation angle vector grid and stacked sparse autoencoder. *Expert Syst.* **2021**, *39*, e12701. [CrossRef]
41. Wagle, S.A.; Harikrishnan, R.; Md Ali, S.H.; Mohammad, F. Classification of Leaves Using New Compact Convolutional Neural Network Models. *Plants* **2022**, *11*, 24._7 [CrossRef] [PubMed]
42. Garbin, C.; Zhu, X.; Marques, O. Dropout vs. batch normalization: An empirical study of their impact to deep learning. *Multimed. Tools Appl.* **2020**, *79*, 12777–12815. [CrossRef]
43. Jaiswal, S.; Nandi, G.C. Robust real-time emotion detection system using CNN architecture. *Neural Comput. Appl.* **2020**, *32*, 11253–11262. [CrossRef]
44. Längkvist, M.; Jendeberg, J.; Thunberg, P.; Loutfi, A.; Lidén, M. Computer aided detection of ureteral stones in thin slice computed tomography volumes using Convolutional Neural Networks. *Comput. Biol. Med.* **2018**, *97* 153–160. [CrossRef] [PubMed]
45. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; pp. 1–14.
46. Sathyanarayana, A.; Joty, S.; Fernandez-Luque, L.; Ofli, F.; Srivastava, J.; Elmagarmid, A.; Arora, T.; Taheri, S. Sleep Quality Prediction From Wearable Data Using Deep Learning. *JMIR mHealth uHealth* **2016**, *4*, e6562.
47. Heydarian, M.; Doyle, T.E.; Samavi, R. MLCM: Multi-Label Confusion Matrix. *IEEE Access* **2022**, *10*, 19083–19095. [CrossRef]
48. Moradi, R.; Berangi, R.; Minaei, B. A survey of regularization strategies for deep models. *Artif. Intell. Rev.* **2020**, *53*, 3947–3986. [CrossRef]

49. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
50. Anandhakrishnan, T.; Jaisakthi, S.M. Identification of tomato leaf disease detection using pretrained deep convolutional neural network models. *Scalable Comput.* **2020**, *21*, 625–635. [CrossRef]
51. Deep Learning, MATLAB2019b. Available online: https://in.mathworks.com (accessed on 30 April 2021)
52. Qiu, W.; Ye, J.; Hu, L.; Yang, J.; Li, Q.; Mo, J.; Yi, W. Distilled-mobilenet model of convolutional neural network simplified structure for plant disease recognition. *Smart Agric.* **2021**, *3*, 109–117. [CrossRef]
53. Awan, M.J.; Rahim, M.S.M.; Salim, N.; Rehman, A.; Nobanee, H.; Shabir, H. Improved Deep Convolutional Neural Network to Classify Osteoarthritis from Anterior Cruciate Ligament Tear Using Magnetic Resonance Imaging. *J. Pers. Med.* **2021**, *11*, 1163. [CrossRef] [PubMed]
54. Durante, M.G.; Rathje, E.M. An exploration of the use of machine learning to predict lateral spreading. *Earthq. Spectra* **2021**, *37*, 2288–2314. [CrossRef]