computers

MDPI

Article

Siamese Network-Based Lightweight Framework for Tomato Leaf Disease Recognition

Selvarajah Thuseethan [1,*], Palanisamy Vigneshwaran [2], Joseph Charles [3] and Chathrie Wimalasooriya [4,*]

[1] Faculty of Science and Technology, Charles Darwin University, Casuarina, NT 0810, Australia
[2] Department of Software Engineering, Sabaragamuwa University of Sri Lanka, Belihuloya 70140, Sri Lanka; vickey@foc.sab.ac.lk
[3] School of Business, University of Southern Queensland, Springfield, QLD 4300, Australia; charles.joseph@unisq.edu.au
[4] School of Computing, University of Otago, Dunedin 9016, Otago, New Zealand
* Correspondence: thuseethan.selvarajah@cdu.edu.au (S.T.); chathrie.wimalasooriya@postgrad.otago.ac.nz (C.W.)

Abstract: In this paper, a novel Siamese network-based lightweight framework is proposed for automatic tomato leaf disease recognition. This framework achieves the highest accuracy of 96.97% on the tomato subset obtained from the PlantVillage dataset and 95.48% on the Taiwan tomato leaf disease dataset. Experimental results further confirm that the proposed framework is effective with imbalanced and small data. The backbone network integrated with this framework is lightweight with approximately 2.9629 million trainable parameters, which is second to SqueezeNet and significantly lower than other lightweight deep networks. Automatic tomato disease recognition from leaf images is vital to avoid crop losses by applying control measures on time. Even though recent deep learning-based tomato disease recognition methods with classical training procedures showed promising recognition results, they demand large labeled data and involve expensive training. The traditional deep learning models proposed for tomato disease recognition also consume high memory and storage because of a high number of parameters. While lightweight networks overcome some of these issues to a certain extent, they continue to show low performance and struggle to handle imbalanced data.

Keywords: plant disease; tomato disease; Siamese network; lightweight; imbalanced data

check for updates

Citation: Thuseethan, S.; Vigneshwaran, P.; Charles, J.; Wimalasooriya, C. Siamese Network-Based Lightweight Framework for Tomato Leaf Disease Recognition. *Computers* **2024**, *13*, 323. https://doi.org/10.3390/computers13120323

Academic Editor: Paolo Bellavista

Received: 1 November 2024
Revised: 29 November 2024
Accepted: 29 November 2024
Published: 4 December 2024

1. Introduction

In the next 30 years, food production needs to be increased approximately by 60% to feed the expected population of 10 billion people on the earth [1]. Various natural and artificial activities significantly reduce food production while anticipating reaching the expected level. The hike in food production with potential losses should meet the global demand for food. In general, plant diseases are a major threat to the modern agricultural industry, heavily reducing the production and food quality [2]. The crop pests and diseases cause a considerable loss of global yields, approximately 21.5% and 17.2% in wheat and potato plants, respectively [3].

Tomato is one of the most economically and nutritionally essential vegetables, which is cultivated all around the globe. The worldwide tomato production has already exceeded 187 million tonnes by 2020, with China being the largest producer, followed by India and the United States of America [4]. Tomato crops can be easily affected by numerous diseases that can cause dramatic economic losses and food shortages. Hence, farmers need to diagnose tomato diseases as early as possible to reduce the risk of losing yields. Apart from preventing tomato diseases by annually testing the garden soil and maintaining an adequate level of potassium, in the past farmers used manual inspection of affected leaves to identify tomato diseases [5]. However, the manual diagnosis of tomato disease is a tedious and time-consuming practice, requiring plant disease-related expertise.

Recently, proposed computational methods to automatically recognize tomato diseases from leaf images used both handcrafted [6] and deep learning [5,7–10] based feature extraction techniques. The handcrafted feature extraction is a process of learning the most informative features to train the classifiers for recognition tasks. The methods adapting the handcrafted feature extraction scheme contain multiple steps in the pipeline, such as pre-processing, feature extraction and classification. Even though handcrafted feature extraction techniques are faster in comparison to deep learning methods, they often show poor performance in unfamiliar conditions. Additionally, the handcrafted techniques are generally not end-to-end trainable, increasing the intricacy of the overall tomato disease recognition system.

On the other hand, very recently proposed deep learning-based image classification techniques showed promising performance across several application domains [11]. Medical, affective computing, social and agriculture are some popular domains where image classification techniques are extensively utilized. Similar to the other image classification tasks, the automated feature extraction-based deep learning techniques have become the benchmarks in tomato disease recognition [5]. Researchers not only utilized the state-of-the-art deep networks [12] but also proposed tailor-made deep networks [7,9] for tomato disease recognition. The majority of the traditional training process of deep networks often demands high computational requirements and large labeled data for training as they use very deep networks. Further, they are not feasible for real-time solutions as they consume a considerable amount of time in both the training and inference phases.

As a potential solution, lightweight deep networks have been widely used for tomato disease recognition, which facilitates smooth mobile deployment and training with small data [13]. However, lightweight network-based tomato disease recognition methods often show limited performance with real-world samples, which is one of the major limitations that hinder its widespread applicability. To enhance the performance, they predominantly rely on additional mechanisms like an attention scheme that further complicate the resultant model [14]. Commonly, lightweight networks with traditional training schemes suffer in obtaining expected classification accuracy with imbalanced data.

To address these issues, in this paper, a novel Siamese network-based lightweight tomato disease recognition framework is proposed. Figure 1 illustrates the overall architecture of the proposed tomato disease recognition framework, which is further explained in Section 3.1. By adopting the key characteristics of lightweight networks and Siamese training, the proposed framework can effectively recognize tomato diseases using imbalanced and small data. Achieving benchmark accuracies with imbalanced and small data is significant for enhancing real-world applicability. The key contributions of this paper are given as follows:

1. A novel Siamese network-based lightweight framework with a contrastive loss is proposed to recognize tomato diseases from small and imbalanced leaf data.
2. A testing procedure dedicated to Siamese networks using majority voting is proposed to enhance the tomato disease recognition accuracy.
3. A set of extensive experiments was carried out on two tomato leaf datasets (https://data.mendeley.com/datasets/ngdgg79rzb/1 (access on 1 January 2024)), including the tomato subset of the PlantVillage dataset and the Taiwan tomato leaves dataset, to show the feasibility of the proposed framework in effectively recognizing diseases.
4. The comparative study shows that the proposed framework can show better performances with small and imbalanced data, in comparison to existing tomato disease recognition from leaf images.

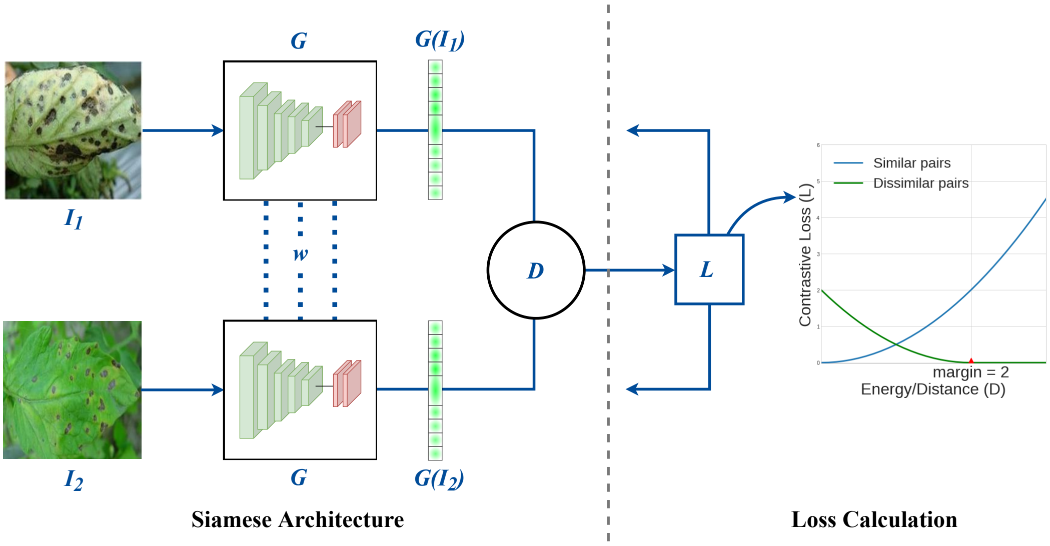

Figure 1. The overall architecture of the proposed tomato disease recognition framework. The I_1 and I_2 are the input images. The weights w are shared between two streams (G) of the Siamese network. Figure 2 illustrates the architecture of the backbone network G. The distance D is estimated as the Euclidean distance between the outputs $G(I_1)$ and $G(I_2)$. The contrastive loss L is then calculated based on D.

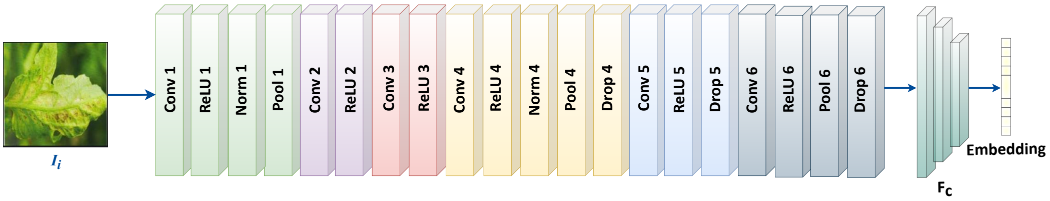

Figure 2. The block diagram of the lightweight deep network used as the backbone of the proposed framework. The layer configuration details are further given in Table 1.

Table 1. Configurations of the backbone CNN architecture integrated with the proposed Siamese network. In this table, w = width, h = height, c = channels, f = number of filters (output channels), ks = kernel size, s = stride, p = padding, sz = amount of neighboring channels, α = multiplicative factor, β = exponent, af = additive factor, pr = probability of an element to be zeroed, i = size of each input and o = size of each output.

Block	Layers	Configuration
0	Input	w = 128, h = 128, c = 3
1	Conv 1	c = 3, f = 64, ks = 5, s = 1, p = 1
	ReLU 1	
	Norm 1	sz = 5, α = 0.0001, β = 0.75, af = 2
	Pool 1	ks = 3, s = 2
2	Conv 2	c = 64, f = 96, ks = 3, s = 1, p = 2
	ReLU 2	
3	Conv 3	c = 96, f = 128, ks = 3, s = 1, p = 2
	ReLU 3	
4	Conv 4	c = 128, f = 96, ks = 3, s = 1, p = 2
	ReLU 4	
	Norm 4	sz = 5, α = 0.0001, β = 0.75, af = 2
	Pool 4	ks = 3, s = 2
	Drop 4	pr = 0.2
5	Conv 5	c = 96, f = 64, ks = 1, s = 1, p = 1
	ReLU 5	
	Drop 5	pr = 0.2

Table 1. *Cont.*

Block	Layers	Configuration
6	Conv 6	$c = 64$, $f = 32$, $ks = 1$, $s = 1$, $p = 1$
	ReLU 6	
	Pool 6	$ks = 3$, $s = 2$
	Drop 6	$pr = 0.2$
7	FC 7	$i = 32 \times 324$, $o = 256$
	ReLU 7	
	Drop 7	$pr = 0.5$
8	FC 8	$i = 256$, $o = 64$
	ReLU 8	
9	FC 9	$i = 64$, $o = 32$

The rest of the paper is organized as follows. A comprehensive analysis of the tomato leaf disease recognition is provided in Section 2. Next, the proposed Siamese network-based framework is presented in Section 3. A set of experimental results is discussed in Section 4. Section 5 concludes the paper with a few future directions on automated tomato leaf disease recognition.

2. Related Work

Diagnosing plant diseases on time in an automatic manner by utilizing the latest information processing techniques has been actively researched lately in the pursuit of supporting the agricultural industry to address its emerging challenges [15]. Over the years, the domain experts in agriculture considered effective recognition of diseases for different crop types, such as citrus [16], apple [17] and pearl millet [18]. Despite the fact that contemporary research has primarily focused on advanced deep learning techniques, in this section, some of the most significant tomato disease recognition techniques covering both traditional and deep learning approaches are reviewed. However, readers are advised to read the recent surveys [19–21], to acquire comprehensive knowledge about more generic plant disease recognition techniques. The comparative study [22], offers insights into the functioning of state-of-the-art deep neural networks in the context of plant disease recognition.

Traditionally, similar to image classification tasks in other domains, handcrafted feature extraction schemes are most widely adopted in tomato disease classification using leaf images [6]. For example, in [6], the color and texture features, along with the kernel mutual subspace method (KMSM), are used for rapid recognition of tomato disease stages. This method achieved an accuracy of 99.34% and 98.66% on a subset of the PlantVillage dataset and AI Challenger 2018 dataset. Further, the proposed method is faster than VGG16 and support vector machine (SVM) in terms of average training and recognition time. Further, various kinds of handcrafted feature extraction techniques have been used for plant disease recognition, in general. However, as the handcrafted feature extraction scheme is a manual process of extracting important features, the methods that use this scheme must follow a series of phases, such as image pre-processing (including segmentation/cropping), feature extraction and classification. This pipeline not only increases the complexity of the tomato disease recognition method but also restricts the end-to-end trainability.

Recent advancements in deep learning for present-day pattern recognition tasks across different domains resulted in creating better opportunities in plant disease recognition [20]. Several deep learning-based methods showed promising results in tomato disease recognition. For example, in an early work, pre-trained AlexNet and VGG16 deep networks were utilized to classify crop diseases from tomato leaves and validated on tomato leaf images taken from the PlantVillage dataset [23]. Both AlexNet and VGG16 deep networks consist of a large number of trainable parameters, and thus they are deemed to be very heavy and demand large training data. Chen et al. [24] proposed a novel Both-channel Residual Attention Network (B-ARNet) model for automated tomato leaf disease recognition. In this study, image denoising and enhancing processes along with background removal were performed prior to extracting the features using the B-ARNet model, which is not ideal for end-to-end system implementation. Unlike the methods discussed so far, a simple tailor-made deep convolutional neural network (CNN) was proposed in [25] for accurate tomato leaf disease detection. However, it showed comparatively poor performance with the lowest accuracy of around 76%. Furthermore, the use of attention schemes has proven to improve the classification accuracy of CNN-based tomato disease recognition models [26]. Attaching additional modules to deep networks often caused increased trainable parameters.

The lightweight deep networks are well known for their reduced memory consumption and real-time applicability. Some lightweight networks like MobileNets provide the facility to find the trade-off between accuracy and latency with the help of tunable hyperparameters [27]. Most recent plant disease recognition methods have aimed at developing lightweight networks because farmers predominantly expect in-the-wild applicability [28]. In [29], a pre-trained MobileNetv2 network was fine-tuned for a three-class tomato disease recognition task. Although a considerable overall accuracy (>90%) was achieved, the proposed method was validated with a simple three-class tomato disease dataset. Bhujel et al. [14] proposed a lightweight model that showed increased overall accuracy and reduced network parameters. In order to enhance the performance of the proposed lightweight model, an attention module was embedded in the original network. One of the key challenges of lightweight architectures was they frequently showed limited performance due to their low generalization capability. Further, the classical training of a network regardless of whether it was heavy or lightweight not only demands a considerable amount of training data but also fails to cope with imbalanced data.

In summary, the majority of the existing tomato leaf disease recognition methods are not feasible to be adopted in real-time settings due to practical problems, such as the unavailability of large and balanced data and the limited computational power and memory. The lightweight Siamese architectures in recent years have shown immaculate performances with small and imbalanced data [30]. Hence, there is a necessity to develop a fully-fledged framework by realizing the advantages of Siamese architecture as the first step toward addressing these challenges.

3. Proposed Method

In this section, the proposed lightweight tomato disease recognition framework using the Siamese network is comprehensively elaborated. First, the overall architecture of the novel Siamese network given in Figure 1 is explained. The proposed architecture is the first one to tackle the imbalanced and small data problems in tomato leaf disease recognition. Second, the loss calculation procedure is elaborated. Third, the newly constructed lightweight CNN that serves as the backbone is given in detail. The potential benefit of this lightweight CNN is it can be trained with small data as it has fewer training parameters. Finally, the proposed novel testing procedure that can generate precise test results is presented.

3.1. Overall Siamese Architecture

Figure 1 provides the overview of the proposed Siamese network-based lightweight tomato leaf disease recognition framework along with the contrastive loss calculation procedure. The in-depth configuration details of the sub-networks (backbone networks) are suppressed for brevity. In general, a Siamese architecture is a special kind of neural network comprising two or more streams of identical backbone networks [30]. In Siamese architectures, both identical sub-networks are constructed with the same parameter sets and network configuration. The proposed Siamese framework consists of two streams of sub-networks that take two different tomato leaf images as inputs at a time.

The output patterns (i.e., feature embeddings) obtained from both sub-networks are compared to find the similarity measure (i.e., the distance), which is then used in the decision-making process. The distance value is based on the pairwise Euclidean distance between the output feature vectors obtained by both streams of the Siamese network. In mathematics, the length of a line segment connecting two points in Euclidean space is referred to as the Euclidean distance, which is the most common distance measure used in machine learning. Ideally, the distance is kept 0 for a similar pair of images and a value > 0 for a dissimilar pair of images. For example, the distance is 0 for both of the images taken from Bacterial spot disease and a value > 0 for one image taken from Bacterial spot disease and the other taken from Black mold disease. During the training phase, for each pair of images, I_1 and I_2, the Euclidean distance D is derived through a parameterized distance function $D(I_1, I_2)$ defined as follows:

$$D(I_1, I_2) = \|G(I_1) - G(I_2)\|_2 \tag{1}$$

where Euclidean distance $D(I_1, I_2)$ is subsequently used as the energy in loss calculation. In the rest of the paper, the Euclidean distance $D(I_1, I_2)$ is referred to as D for conciseness.

3.2. Loss Calculation

In this method, a widely popular contrastive loss proposed in [31] is exploited. The contrastive loss calculation is performed over the pairs of input samples, in contrast to the conventional method, where the loss is calculated as the sum over input samples. In the contrastive loss, the similar input pairs are pushed close by while the dissimilar input samples are pulled apart. The contrastive loss calculation for the *ith* pair of input images is given in Equation (2).

$$L(Y, [I_1, I_2]^i) = (1 - Y)L_s(D^i) + (Y)L_d(D^i) \tag{2}$$

where, $L(Y, [I_1, I_2]^i)$ is the loss for the *ith* image pair $[I_1, I_2]^i$ with the label Y. The partial loss for similar pairs L_s and dissimilar pairs L_d is designed in such a way that the overall loss L is minimized. Correspondingly, the D fetch a small value for similar input pairs and a large value for dissimilar input pairs. The resultant contrastive loss is given below.

$$L = \underbrace{(1 - Y)\frac{1}{2}(D)^2}_{\text{similar pair loss}} + \underbrace{(Y)\frac{1}{2}\{max(0, m - D)\}^2}_{\text{dissimilar pair loss}} \tag{3}$$

here, Y indicates a binary label given to an input image pair $[I_1, I_2]$. For similar and dissimilar input pairs, Y has the values 0 and 1, respectively. m is the margin set to a value greater than 0. For dissimilar pairs, any distance D within the margin m (i.e., $m - D > 0$) is penalized. In this case, the margin m is set to 2 and defines the loss tendency of dissimilar pairs, as shown in the loss calculation fragment of Figure 1. Likewise, similar pairs contribute to the contrastive loss if the distance $D > 0$.

3.3. Lightweight Backbone Network

In recent years, deep networks with fewer layers have become very popular in the image classification domain due to their lightweight nature. As the primary focus of this study is to develop the Siamese framework lighter, the number of layers is retained to a small value, and so the trainable parameters of this network are $\approx$ 2.96 million. As described in Figure 2 and Table 1, a nine-block deep CNN is proposed as the backbone network. The in-depth details about the network configurations of the backbone CNN architecture are provided in Table 1. This model architecture is designed purely based on a heuristic way, where both empirical results and a trial-and-error approach are utilized. The layer configurations are also continuously changed until the best performance is achieved.

The images of the size $3 \times 128 \times 128$ (i.e., channel $\times$ width $\times$ height) can be fed as the inputs, as seen in Table 1. This network consists of six convolution layers and three connected layers to learn the most important feature representations. The receptive fields of the convolution layers are kept very small (5×5, 3×3 and 1×1). More importantly, the receptive field of 1 × 1 used in convolution layers 5 and 6 portrays the linear transformation function that improves the non-linear property of the neural network. The rectified linear (ReLU) layers are stacked after each convolution layer (i.e., convolution layers 1-6) and after the first two fully connected layers (i.e., fully connected layers 7 and 8), as they also consistently increase the non-linearity.

The local response normalization (LRM) function, referred to as Norm 1 and 4 in Table 1, stimulates lateral inhibition that performs local contrast enhancement. Two LRM layers are appropriately affixed in blocks 1 and 4 immediately after ReLU layers. Further, two-dimensional max-pooling layers (indicated as Pool in Table 1) that estimate each patch's maximum value in every feature map are placed in blocks 1, 4 and 6. In blocks 4, 5, 6 and 7, three dropout layers with probability retention of 0.2, 0.2, 0.2 and 0.5 are stacked, respectively. The final layer of the backbone CNN architecture is a fully connected layer with 32 neurons.

3.4. Majority Voting-Based Testing Scheme

In order to obtain the testing accuracy of the proposed tomato disease recognition approach, a querying mechanism involving support images and the query image is exploited, as illustrated in Figure 3. In most Siamese network-based architectures, the testing is performed with the help of precisely one support image. This testing scheme has the advantage of calculating the test accuracy using a majority voting with five support images, which can alleviate the problems in test accuracy created by accidental causes.

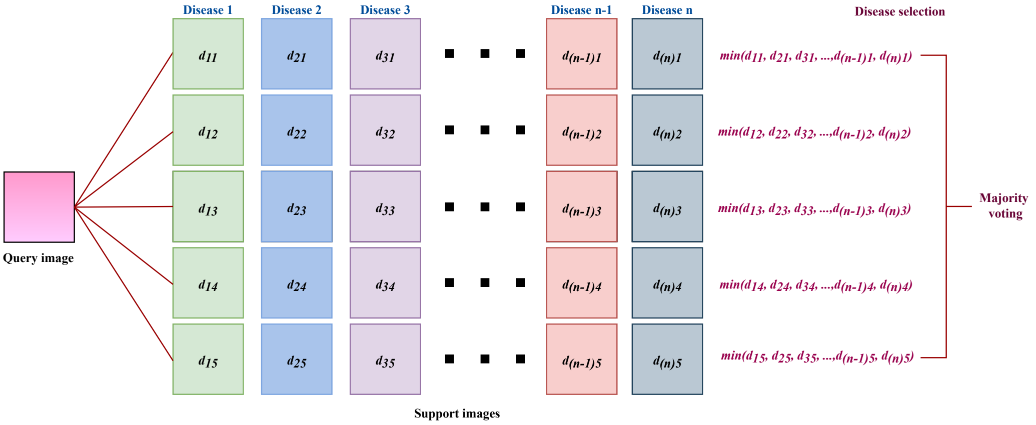

Figure 3. A proposed testing scheme dedicated to Siamese network-based tomato plant disease recognition. This scheme follows a majority voting mechanism.

Consider that there are n tomato disease types ($Disease$ 1 to n) to be classified. For each disease, five support images are arbitrarily selected. The distance $d_{i,j}$ is estimated between the query image and the jth support image of disease i, where j takes 1 to 5 and i takes 1 to n. The disease type of the image with minimum distance amongst a row of support images is selected as the predicted disease class, which is given by the following:

$$Disease^j = f_{disease}(\min_{i=1...n}(d_{ij})) \tag{4}$$

where, j is a constant that will take the values between 1 and 5 for multiple runs of this equation. The $f_{disease}(\cdot)$ is a function that returns the disease type for the input given. For example, the $Disease^1 = Disease(n-1)$ if the $d_{(n-1)1}$ shows the minimum distance for $j = 1$.

Once the $Disease^j$ for all $j = 1 \ldots 5$ is decided, a majority voting scheme is applied to make the final decision. To perform this, first, the occurrences of each tomato leaf disease type selected within $Disease^j$ are counted. Next, the tomato disease type with the highest occurrence count is selected as the predicted tomato disease class. For instance, the tomato leaf $Disease3$ can be fixed as the predicted class if it is selected three out of five times. In the case of two or more equal number of maximum occurrences, the average distances of the disease types with maximum occurrences are used as the deciding factor. Precisely, the predicted tomato disease type is selected as the one with a minimum average distance from the maximum occurrents.

4. Experiments

In this section, the results obtained from a set of experiments performed in various environments are discussed in detail. The primary focus of these experiments is to demonstrate the feasibility of the proposed model with small and imbalanced data. First, the tomato disease datasets are described, followed by implementation details. Next, the results are discussed in detail, along with the comparison of existing state-of-the-art methods.

4.1. Datasets

To evaluate the proposed lightweight tomato leaf disease recognition framework, two benchmark datasets, such as a subset of the PlantVillage dataset containing tomato leaf disease samples and the Taiwan tomato disease dataset, are used [32]. Sample images picked from each class of both datasets are given in Figure 4.

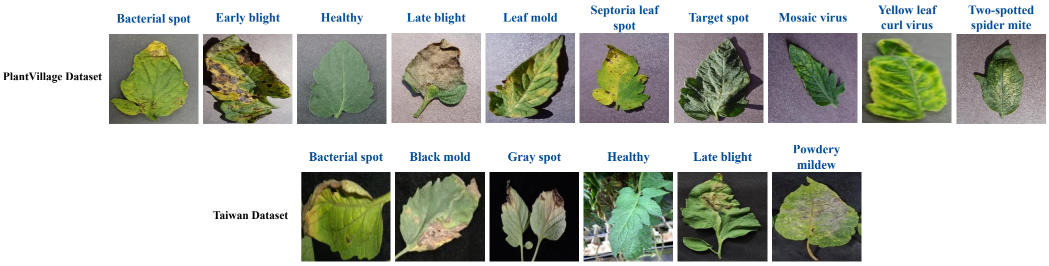

Figure 4. Sample images taken from the PlantVillage and Taiwan datasets for each tomato disease class.

The *PlantVillage tomato subset* consists of 14,531 image samples for ten classes, including nine tomato leaf disease classes and one healthy class. The tomato leaf diseases included in this dataset are Bacterial spot (1702), Early blight (800), Late blight (1528), Leaf mold (762), Septoria leaf spot (1417), Target Spot (1124), Mosaic virus (299), Yellow leaf curl virus (4286) and Two-spotted spider mite (1341). This is a highly imbalanced dataset with an uneven distribution of image samples, where the majority class (4286 samples of Yellow leaf curl virus disease class) consists of a fourteen times (14×) larger sample size than the minority class (299 of Mosaic virus disease class). The image size of the original dataset is 227 × 227.

The original *Taiwan tomato disease dataset* is relatively small and used to evaluate the proposed model, which has 622 leaf samples of the size 227 × 227 for six categories, including healthy and diseased classes. Bacterial spot (100), Black mold (67), Gray spot (84), Healthy (106), Late blight (98) and Powdery mildew (157) are the classes included in this dataset. Furthermore, there is an enhanced dataset containing more samples for each class that are generated using a range of augmentation functions, such as clockwise rotation (by 90, 180 and 270 degrees), mirroring (horizontally and vertically) and changing image brightness, which is also exploited. Figure 5 provides an example image with all the augmented samples. There are 4976 tomato leaf images in the final dataset with augmented samples.

4.2. Implementations and Evaluation Metrics

The proposed framework and the other compared methods are implemented using the PyTorch (https://pytorch.org/ (access on 1 January 2024)) open source machine learning framework and trained on Google Colab (https://colab.research.google.com/ (access on 1 January 2024)) notebook. The models are evaluated on Intel(R) Xeon(R) CPU @ 2.30 GHz supported by NVIDIA Tesla T4 GPU with 13 GB RAM.

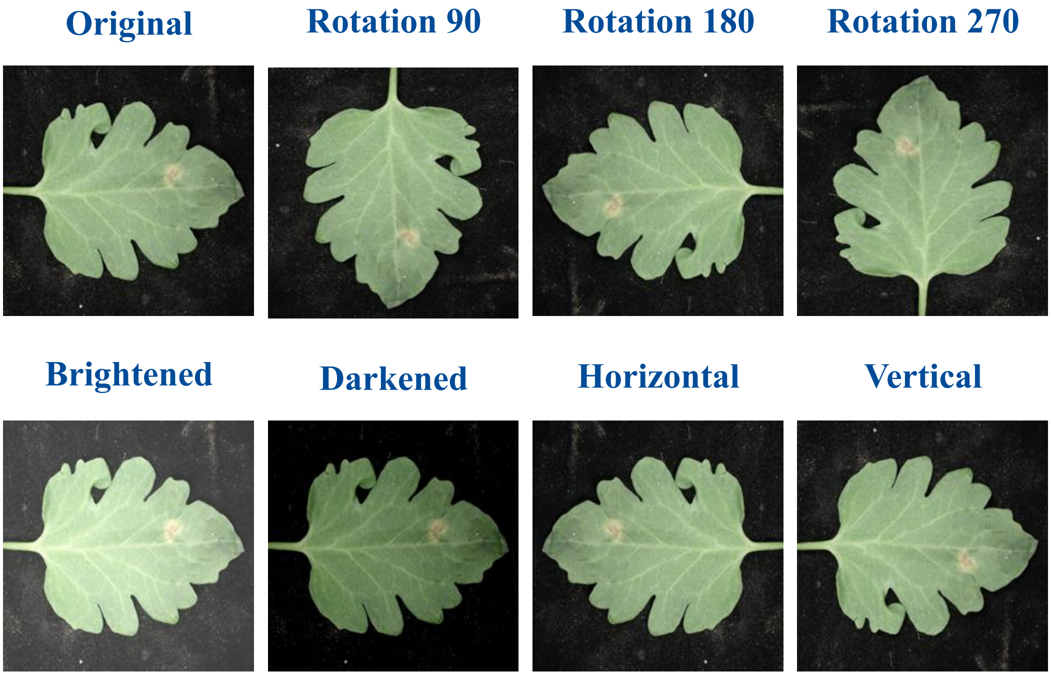

Figure 5. An example image taken from Black mold disease and its augmented samples. There are seven different augmentation functions applied to the leaf images.

All the models are trained for 10 epochs with a batch size of 8. In the training phase, the initial learning rate of the Adam optimizer is set to 0.001 with a weight decay of 0.0001. Further, the results on the Taiwan tomato disease dataset are recorded for the dedicated test set provided by the authors, while the results on the PlantVillage tomato subset are presented using a 10-fold cross-validation technique. To present the experimental results, a popular evaluation metric, average accuracy, is used. The closeness of the predicted tomato disease class to the true tomato disease class can be effectively described through accuracy. The equation to obtain accuracy is defined below.

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{5}$$

where, t_p is true positive, t_n is true negative, f_p is false positive and f_n is false negative.

4.3. Results

The testing framework illustrated in Figure 3 is exploited to obtain the testing accuracy of the proposed tomato disease recognition approach. Figure 6 provides an example test scenario where a Gray spot test image is queried against the support images. The query image obtained smaller distances with all five Gray spot support images compared to the query images of other classes. Hence, using the majority voting scheme, the query image is correctly classified as the Gray spot diseased leaf on this occasion.

Figure 7 presents the loss observed for the model trained with 100% of the data from the PlantVillage tomato leaf dataset. The loss curve demonstrates that the model learned smoothly with the given parameters. Table 2 summarizes the overall accuracy obtained for the proposed framework on the PlantVillage and both variations of the Taiwan tomato leaf disease datasets. The results are presented for the models trained with 100%, 75% and 50% training data. The proposed framework trained with the 100% training set obtained an accuracy of 95.48%, 96.14% and 96.97% on the original Taiwan dataset, the Taiwan dataset with augmentation and the PlantVillage dataset, respectively. In all three cases, an increase in accuracy is noted for the Taiwan dataset with augmentation compared to the original Taiwan dataset, which confirms that the augmentation enhances the performance of the proposed framework. Further, the overall accuracy is consistent even after reducing the training data to 75% and 50%, with a maximum 4% accuracy drop recorded for the Taiwan dataset. For the PlantVillage tomato leaf dataset, only a 1.5% drop is witnessed when reducing the training set by 50%; it is convincing that the proposed framework can show good performance with small data.

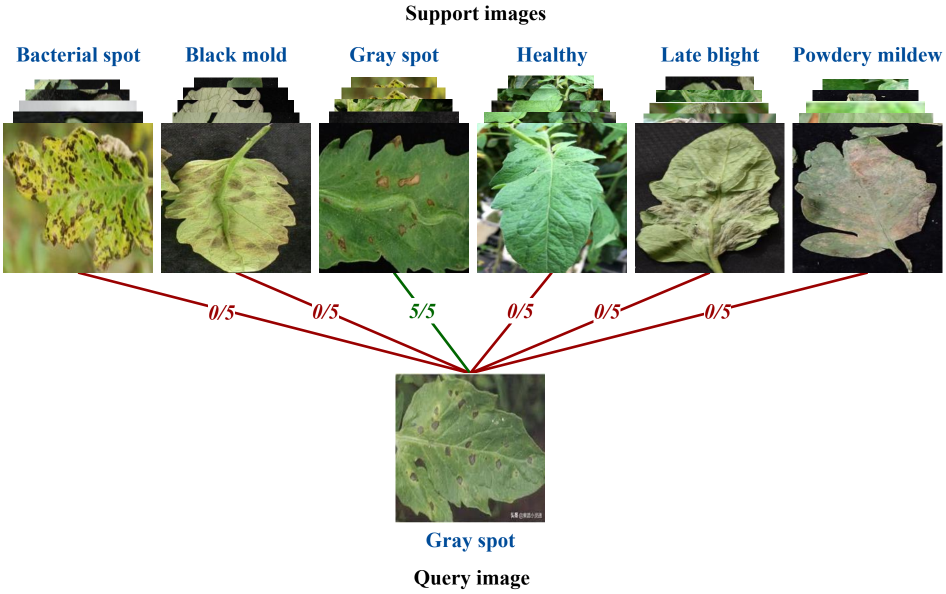

Figure 6. Querying a test sample from the Gray spot disease class of the Taiwan dataset.

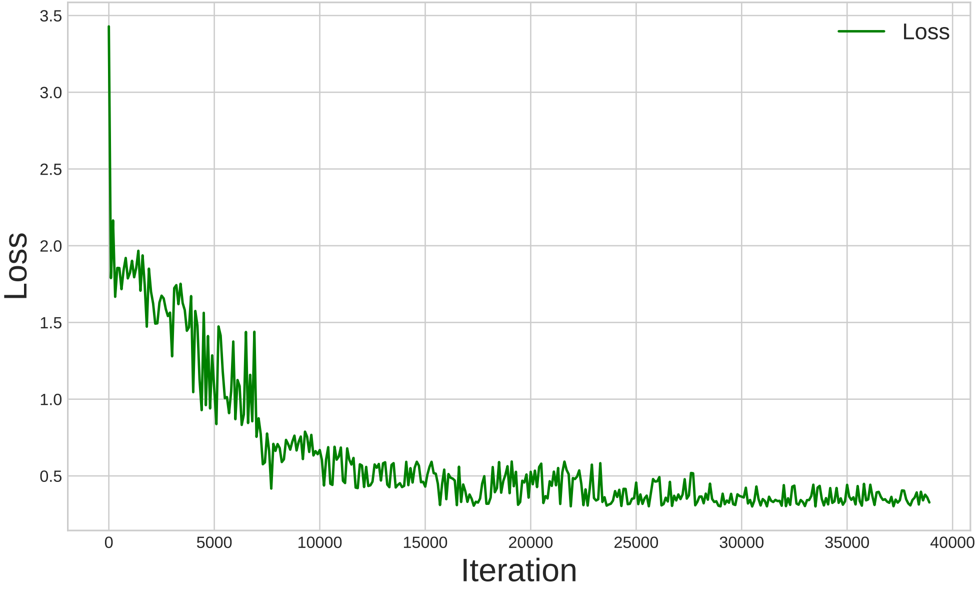

Figure 7. The loss observed for the model trained on the whole PlantVillage tomato leaf dataset.

Figure 8 shows the number of samples and accuracies obtained for each class in the PlantVillage dataset. The results are recorded for the model trained on 100% data. In each pair of bars, the left and right bars indicate the number of samples and the class-wise accuracies, respectively. As can be seen, the tomato subset of the PlantVillage dataset is highly imbalanced, where the Mosaic virus disease is the minority class with 299 samples and the Yellow leaf curl virus is the majority class with 4286 samples (i.e., greater than 14× of the minority class). However, the proposed framework obtained higher class-wise accuracies for minority classes. This demonstrates that the proposed framework can consistently perform well on imbalanced data.

Table 2. Average accuracy achieved on the benchmark tomato leaf disease datasets. The results are separately recorded for the models trained with 50%, 75% and 100% of the training set.

Databases	**Training Data**		
	100%	**75%**	**50%**
Taiwan	0.9548	0.9262	0.9103
Taiwan/w augmentation	0.9614	0.9434	0.9279
PlantVillage	0.9697	0.9609	0.9546

The accuracy presented for both variations of the Taiwan dataset is based on the dedicated test data provided by the authors.

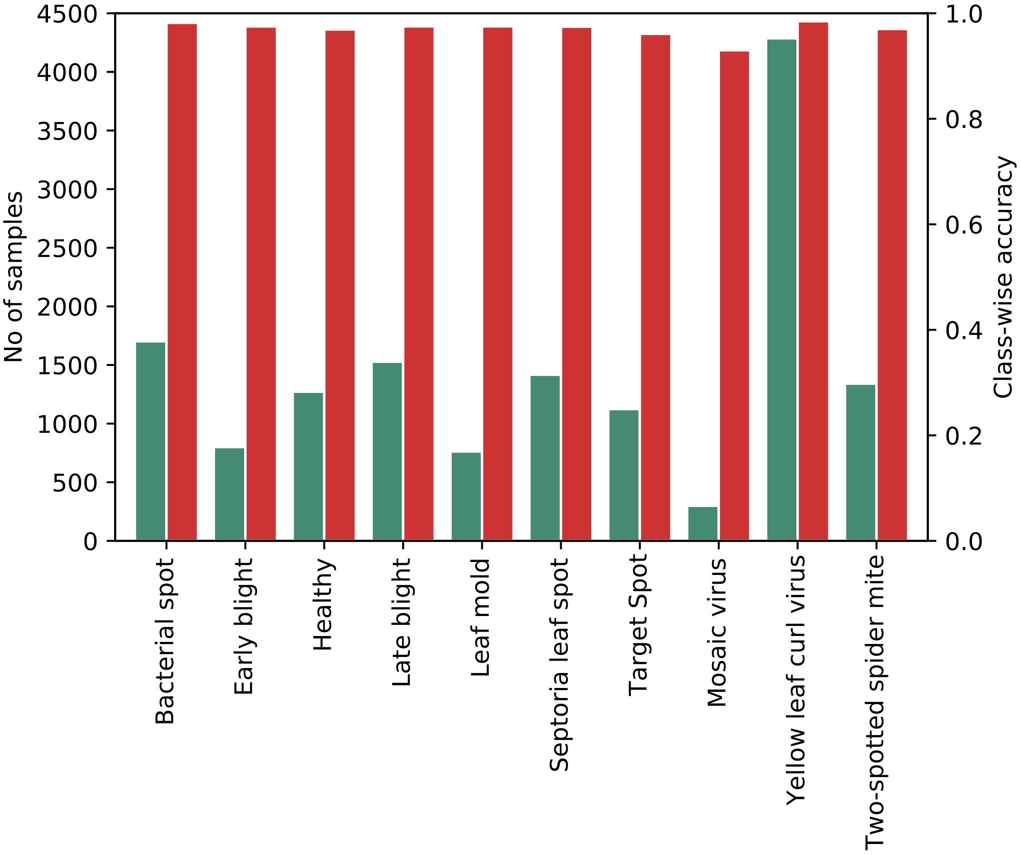

Figure 8. Performance evaluation with imbalanced data: per class samples and class-wise accuracies for the PlantVillage dataset are compared.

4.4. Comparison

Two types of comparisons are performed to show the superiority of the proposed framework. In the first case, the backbone network of the proposed framework is replaced by other state-of-the-art networks. The rest of the procedures are kept identical, including the training and testing schemes. In the second case, some of the recently proposed tomato disease recognition methods are compared. In this case, the same training and testing environments are maintained for a fair evaluation.

Table 3 provides the comparison results obtained for the proposed lightweight network and 11 state-of-the-art networks, as the backbone of the proposed framework. In this comparison, both the heavy networks and lightweight networks are included. In order to perform a fair evaluation, all the backbones are trained from scratch. The proposed framework with the novel lightweight deep network achieved the best accuracy on all tomato datasets with 100%, 75% and 50% data. In particular, the proposed framework with a novel lightweight deep network showed only a slight depreciation in the accuracy when the data were reduced to 75% and 50%. This is a promising feature as the performance of most of the existing state-of-the-art networks drops dramatically while reducing the training data. Notably, another lightweight architecture MobileNetv2, also consistently performed across all datasets with the second-best accuracy. However, its accuracy was significantly lower than the proposed framework integrated with the novel lightweight network.

Table 3. Comparison of the average accuracies obtained on the PlantVillage, Taiwan and Taiwan with augmentation tomato leaf disease datasets. The trainable parameters for each single stream network are also compared.

Deep Networks	Params	PlantVillage			Taiwan			Taiwan/w Augmentation		
		100%	75%	50%	100%	75%	50%	100%	75%	50%
AlexNet	61.1008	0.7128	0.6812	0.6137	0.6896	0.6039	0.5527	0.7064	0.6788	0.6438
VGG19	143.6672	0.7378	0.7031	0.6374	0.6931	0.6438	0.6078	0.7217	0.7033	0.6612
ResNet152	60.1928	0.8316	0.7812	0.7079	0.6018	0.5178	0.4538	0.7423	0.7164	0.6891
DenseNet201	20.0139	0.8134	0.7521	0.7215	0.6135	0.5730	0.5137	0.7239	0.6433	0.6138
Inceptionv3	27.1613	0.8436	0.8127	0.7513	0.6847	0.6037	0.5678	0.7869	0.7519	0.7238
SqeezeNet1_1	1.2355	0.7825	0.7436	0.7212	0.6622	0.6231	0.5972	0.7634	0.7439	0.7094
ShuffleNetv2	7.3940	0.7563	0.7103	0.6479	0.7231	0.6012	0.5678	0.7013	0.6833	0.6671
MobileNetv2	3.5049	0.9103	0.8563	0.8117	0.8236	0.7863	0.7631	0.8437	0.8032	0.7769
Ours	2.9629	**0.9697**	**0.9609**	**0.9546**	**0.9548**	**0.9262**	**0.9103**	**0.9614**	**0.9434**	**0.9279**

The accuracy presented for both variations of the Taiwan dataset is based on the dedicated test data provided by the authors.

The trainable parameters of each backbone network are compared in Table 3. The proposed lightweight network has nearly 2.9629 million trainable parameters, which is less than many state-of-the-art deep networks. SqueezeNet has the lowest trainable parameters among the competitors, with approximately 1.2355 million. Another lightweight network, namely ShuffleNet, has around 7.3940 million parameters. Except for these three lightweight networks, all the other networks are very deep networks with larger trainable parameters. For instance, the VGG19 has nearly 143,6672 parameters to tune during the training, which is time-consuming and requires large data. The proposed framework is faster and shows high performance with small data because the novel lightweight network integrated into the proposed framework has few parameters.

A comparison between the proposed framework and other existing tomato leaf disease recognition methods is given in Table 4. The results presented for the compared methods are not directly taken from the respective papers as the experimental environment is entirely different. Hence, these methods are rerun on the tomato datasets under the same environment with 100%, 75% and 50% data for comparison. As can be seen from the comparison, the proposed framework comprehensively outperformed existing tomato leaf disease recognition methods. The proposed framework showed a 13.16%, 25.33% and 14.05% accuracy improvement over its competitor [33], with 50% data on the PlantVillage, Taiwan and Taiwan with augmentation tomato datasets, respectively. The proposed framework showed a higher accuracy improvement (25.33%) with the Taiwan dataset, which demonstrates that the proposed framework can better recognize tomato leaf disease recognition with small data.

Table 4. Comparison of existing tomato leaf disease recognition methods.

Methods	PlantVillage			Taiwan			Taiwan/w Augmentation		
	100%	75%	50%	100%	75%	50%	100%	75%	50%
GoogLeNet and VGG16 [25]	0.8863	0.8136	0.7125	0.6821	0.6338	0.5822	0.8368	0.7791	0.7137
ABCK-BWTR and B-ARNet [24]	0.9012	0.8633	0.7473	0.8036	0.7191	0.6361	0.8672	0.7736	0.7236
CNN [34]	0.8463	0.7966	0.7138	0.7120	0.6731	0.6133	0.7463	0.7038	0.6477
Deep CNN [14]	0.9136	0.8817	0.7968	0.7934	0.7367	0.6982	0.8892	0.8367	0.7962
RF, LR, SVM, KNN [33]	0.9321	0.9068	0.8436	0.8463	0.7792	0.7263	0.9132	0.8766	0.8136
Ours	**0.9697**	**0.9609**	**0.9546**	**0.9548**	**0.9262**	**0.9103**	**0.9614**	**0.9434**	**0.9279**

The accuracy presented for both variations of the Taiwan dataset is based on the dedicated test data provided by the authors.

5. Conclusions

Timely recognition of tomato leaf diseases with small and imbalanced data is essential to overcome practical challenges related to tomato crop cultivation, such as unprecedented losses. So far, only a few existing methods have studied tomato leaf disease recognition. However, none of them have explored the small and imbalanced data problems in this domain. It is also important to develop a lightweight deep-learning approach that can be deployed directly on resource-constrained devices. Consequently, our proposed framework's lightweight nature ensures compatibility with such devices. The preliminary investigation confirmed that it can run effectively on devices with at least 2 GB RAM and a CPU architecture equivalent to a 64-bit processor or higher. Hence, in this work, a Siamese network-based lightweight framework is proposed to perform effective tomato leaf disease recognition.

The proposed framework is integrated with a novel lightweight deep network as the backbone. The proposed framework, along with the novel testing procedure, is tested on three benchmark tomato leaf datasets, namely the PlantVillage, Taiwan and Taiwan with augmentation. The results demonstrated that the proposed framework effectively recognizes tomato diseases with small and imbalanced training data. In addition, during the study, the overfitting was controlled by using data augmentation and dropout layers. Furthermore, the proposed framework's single-stream network consists of 2.9629 million trainable parameters, which is significantly fewer than the existing state-of-the-art deep networks. This reduction in processing power makes it a better approach compared to other studies and represents a significant contribution to this research. Comparative analysis confirms that the proposed tomato leaf disease recognition framework is superior to recently proposed methods mentioned on the Table 4, achieving accuracies of 96.97% and 95.48% on the tomato dataset obtained from PlantVillage and Taiwan tomato leaf disease, respectively.

In the future, the proposed framework will be evaluated with the data collected in the wild as it is significant to test the in-field applicability.

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

Funding: This research received no external funding.

Data Availability Statement: https://data.mendeley.com/datasets/ngdgg79rzb/1 (access on 1 January 2024).

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

References

1. FAO. *The State of Food and Agriculture 2019. Moving Forward on Food Loss and Waste Reduction*; FAO: Rome, Italy, 2019; pp. 2–13.
2. Strange, R.N.; Scott, P.R. Plant Disease: A Threat to Global Food Security. *Annu. Rev. Phytopathol.* **2005**, *43*, 83–116. [CrossRef]
3. Savary, S.; Willocquet, L.; Pethybridge, S.J.; Esker, P.; McRoberts, N.; Nelson, A. The global burden of pathogens and pests on major food crops. *Nat. Ecol. Evol.* **2019**, *3*, 430–439. [CrossRef] [PubMed]
4. Panno, S.; Davino, S.; Caruso, A.G.; Bertacca, S.; Crnogorac, A.; Mandić, A.; Noris, E.; Matić, S. A review of the most common and economically important diseases that undermine the cultivation of tomato crop in the mediterranean basin. *Agronomy* **2021**, *11*, 2188. [CrossRef]
5. Eligar, V.; Patil, U.; Mudenagudi, U. Performance Analysis of Deep Learning Algorithms Toward Disease Detection: Tomato and Potato Plant as Use-Cases. In *Intelligent and Cloud Computing*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 595–606.
6. Zhang, Y.; Wu, H.; Chen, C.; Wang, X. Rapid Recognition of Tomato's Disease Stages Based on the Kernel Mutual Subspace Method. *Appl. Eng. Agric.* **2021**, *37*, 793–804. [CrossRef]
7. Kim, S.K.; Ahn, J.G. Tomato Crop Diseases Classification Models Using Deep CNN-based Architectures. *J. Korea-Acad.-Ind. Coop. Soc.* **2021**, *22*, 7–14.
8. Liu, J.; Wang, X. Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model. *Plant Methods* **2020**, *16*, 1–16. [CrossRef]
9. Yoren, A.I.; Suyanto, S. Tomato Plant Disease Identification through Leaf Image using Convolutional Neural Network. In Proceedings of the 2021 9th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia, 3–5 August 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 320–325.
10. Zhang, Y.; Song, C.; Zhang, D. Deep learning-based object detection improvement for tomato disease. *IEEE Access* **2020**, *8*, 56607–56614. [CrossRef]
11. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
12. Chen, H.C.; Widodo, A.M.; Wisnujati, A.; Rahaman, M.; Lin, J.C.W.; Chen, L.; Weng, C.E. AlexNet convolutional neural network for disease detection and classification of tomato leaf. *Electronics* **2022**, *11*, 951. [CrossRef]
13. Wu, Y.; Xu, L. Lightweight compressed depth neural network for tomato disease diagnosis. In Proceedings of the Eleventh International Conference on Graphics and Image Processing (ICGIP 2019), Hangzhou, China, 12–14 October 2019; SPIE: Bellingham, WA, USA, 2020; Volume 11373, pp. 491–499.
14. Bhujel, A.; Kim, N.E.; Arulmozhi, E.; Basak, J.K.; Kim, H.T. A lightweight Attention-based convolutional neural networks for tomato leaf disease classification. *Agriculture* **2022**, *12*, 228. [CrossRef]
15. Ngugi, L.C.; Abelwahab, M.; Abo-Zahhad, M. Recent advances in image processing techniques for automated leaf pest and disease recognition–A review. *Inf. Process. Agric.* **2021**, *8*, 27–51. [CrossRef]
16. Janarthan, S.; Thuseethan, S.; Rajasegarar, S.; Lyu, Q.; Zheng, Y.; Yearwood, J. Deep metric learning based citrus disease classification with sparse data. *IEEE Access* **2020**, *8*, 162588–162600. [CrossRef]
17. Yu, H.; Cheng, X.; Chen, C.; Heidari, A.A.; Liu, J.; Cai, Z.; Chen, H. Apple leaf disease recognition method with improved residual network. *Multimed. Tools Appl.* **2022**, *81*, 7759–7782. [CrossRef]
18. Kundu, N.; Rani, G.; Dhaka, V.S.; Gupta, K.; Nayak, S.C.; Verma, S.; Ijaz, M.F.; Woźniak, M. IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* **2021**, *21*, 5386. [CrossRef]
19. 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]
20. Li, L.; Zhang, S.; Wang, B. Plant disease detection and classification by deep learning—A review. *IEEE Access* **2021**, *9*, 56683–56698. [CrossRef]
21. Dhaka, V.S.; Meena, S.V.; Rani, G.; Sinwar, D.; Ijaz, M.F.; Woźniak, M. A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors* **2021**, *21*, 4749. [CrossRef]
22. George, R.; Thuseethan, S.; Ragel, R.G. Comparative Analysis of Pre-trained Deep Neural Networks for Plant Disease Classification. In Proceedings of the 2024 21st International Joint Conference on Computer Science and Software Engineering (JCSSE), Phuket, Thailand, 19–22 June 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 179–186.
23. Rangarajan, A.K.; Purushothaman, R.; Ramesh, A. Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia Comput. Sci.* **2018**, *133*, 1040–1047. [CrossRef]
24. Chen, X.; Zhou, G.; Chen, A.; Yi, J.; Zhang, W.; Hu, Y. Identification of tomato leaf diseases based on combination of ABCK-BWTR and B-ARNet. *Comput. Electron. Agric.* **2020**, *178*, 105730. [CrossRef]
25. Agarwal, M.; Singh, A.; Arjaria, S.; Sinha, A.; Gupta, S. ToLeD: Tomato leaf disease detection using convolution neural network. *Procedia Comput. Sci.* **2020**, *167*, 293–301. [CrossRef]
26. Zhao, S.; Peng, Y.; Liu, J.; Wu, S. Tomato leaf disease diagnosis based on improved convolution neural network by attention module. *Agriculture* **2021**, *11*, 651. [CrossRef]
27. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
28. Thakur, P.S.; Khanna, P.; Sheorey, T.; Ojha, A. Trends in vision-based machine learning techniques for plant disease identification: A systematic review. *Expert Syst. Appl.* **2022**, *208*, 118117. [CrossRef]

29. Zaki, S.Z.M.; Zulkifley, M.A.; Stofa, M.M.; Kamari, N.A.M.; Mohamed, N.A. Classification of tomato leaf diseases using MobileNet v2. *Iaes Int. J. Artif. Intell.* **2020**, *9*, 290. [CrossRef]
30. Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; Shah, R. Signature verification using a "siamese" time delay neural network. In Proceedings of the 6th International Conference on Neural Information Processing Systems, Denver, CO, USA, 29 November–2 December 1993; Volume 6.
31. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; IEEE: Piscataway, NJ, USA, 2006; Volume 2, pp. 1735–1742.
32. Huang, M.L.; Chang, Y.H. Dataset of tomato leaves. *Mendeley Data* **2020**, *1*. [CrossRef]
33. Astani, M.; Hasheminejad, M.; Vaghefi, M. A diverse ensemble classifier for tomato disease recognition. *Comput. Electron. Agric.* **2022**, *198*, 107054. [CrossRef]
34. Trivedi, N.K.; Gautam, V.; Anand, A.; Aljahdali, H.M.; Villar, S.G.; Anand, D.; Goyal, N.; Kadry, S. Early detection and classification of tomato leaf disease using high-performance deep neural network. *Sensors* **2021**, *21*, 7987. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.