

Article

# Monitoring Tomato Leaf Disease through Convolutional Neural Networks

Antonio Guerrero-Ibañez <sup>1</sup>  and Angelica Reyes-Muñoz <sup>2,\*</sup><sup>1</sup> Faculty of Telematics, University of Colima, Colima 28040, Mexico<sup>2</sup> Computer Architecture Department, Polytechnic University of Catalonia, 08860 Barcelona, Spain

\* Correspondence: angelica.reyes@upc.edu

**Abstract:** Agriculture plays an essential role in Mexico's economy. The agricultural sector has a 2.5% share of Mexico's gross domestic product. Specifically, tomatoes have become the country's most exported agricultural product. That is why there is an increasing need to improve crop yields. One of the elements that can considerably affect crop productivity is diseases caused by agents such as bacteria, fungi, and viruses. However, the process of disease identification can be costly and, in many cases, time-consuming. Deep learning techniques have begun to be applied in the process of plant disease identification with promising results. In this paper, we propose a model based on convolutional neural networks to identify and classify tomato leaf diseases using a public dataset and complementing it with other photographs taken in the fields of the country. To avoid overfitting, generative adversarial networks were used to generate samples with the same characteristics as the training data. The results show that the proposed model achieves a high performance in the process of detection and classification of diseases in tomato leaves: the accuracy achieved is greater than 99% in both the training dataset and the test dataset.

**Keywords:** convolutional neural networks; deep learning; disease classification; generative adversarial network; tomato leaf



**Citation:** Guerrero-Ibañez, A.; Reyes-Muñoz, A. Monitoring Tomato Leaf Disease through Convolutional Neural Networks. *Electronics* **2023**, *12*, 229. <https://doi.org/10.3390/electronics12010229>

Academic Editors: Taiyong Li, Wu Deng and Jiang Wu

Received: 8 November 2022

Revised: 20 December 2022

Accepted: 20 December 2022

Published: 2 January 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

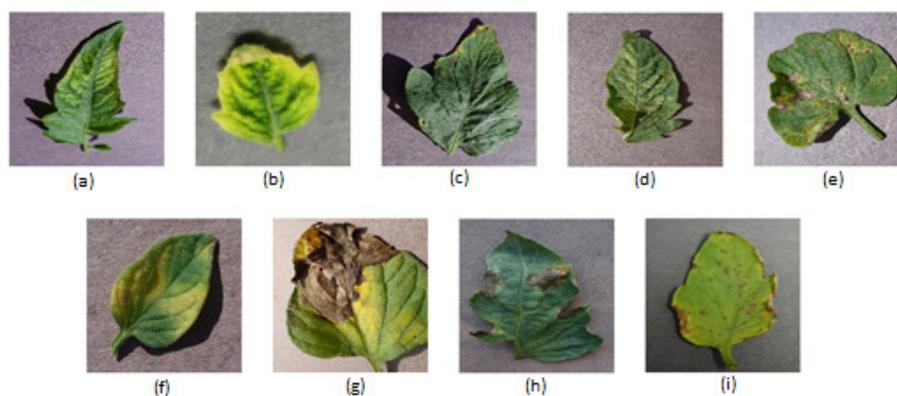
## 1. Introduction

Tomato is one of the most common vegetables grown worldwide and is a high source of income for farmers. The 2020 statistical report of the Food and Agriculture Organization Corporate Statistical Database (FAOSTAT) indicates that world tomato production was 186.821 million tons [1]. In Mexico, the tomato is one of the main crops within the national production, being considered as a basic ingredient both in Mexican cuisine and in general in the cuisine of various parts of the world. According to a report published by Our World in Data in 2020, Mexico is among the top ten countries with the highest production of tomatoes, with a production of 4.1 million tons per year [2]. The Mexican Ministry of Agriculture, Livestock, Rural Development, Fishing and Food (MALRDFF) through the AgriFood and Fisheries Information Service presented the report on Mexico's AgriFood Trade Balance, indicating that tomato is the second most exported agricultural product, with avocado taking the first place. Besides this, tomato production in Mexico has an annual variation of 5.3% from 2011 to 2020 [3]. However, production is affected by different circumstances. The Food and Agriculture Organization (FAO) estimates that crop diseases are responsible for losses ranging from 20 to 40% of total production [4]. Various diseases of the tomato plant can affect the product in terms of quantity and quality, thus decreasing productivity. Diseases can be classified into two main groups [5]. The first group of diseases is related to infectious microorganisms including viruses, bacteria, and fungi. These types of diseases can spread rapidly from plant to plant in the field when environmental conditions are favorable. The second group of diseases is caused by non-infectious chemical or physical factors including adverse environmental factors, physiological or nutritional disorders and

herbicide injury. While it is true that non-infectious diseases cannot spread from plant to plant, diseases can spread if the entire plantation is exposed to the same adverse factor [6].

Some special conditions can cause plant diseases. Specifically, there is a conceptual model known as the disease triangle which describes the relationship between three essential factors: the environment, the host and the infectious agent. If any of these three factors is not present, then the triangle is incomplete, and therefore the disease does not occur. There are abiotic factors such as air flow, temperature, humidity, pH, and watering that can significantly affect the plant. The infectious agent is a kind of organism that attacks the plant such as fungi, bacteria, virus, among others. The host is the plant which is affected by a pathogen. When these factors occur simultaneously, disease is produced [7]. Generally, diseases are manifested by symptoms that affect the plant from the bottom up and many of these diseases have a rapid spread process after infection.

Figure 1 shows some of the most common diseases affecting tomato leaves including mosaic virus, yellow leaf curl virus, target spot, two-spotted spider mite, septoria leaf spot, leaf mold, late blight, early blight, and bacterial spot.



**Figure 1.** Representative images of the most common diseases affecting tomato leaves: (a) mosaic virus, (b) yellow leaf curl virus, (c) target spot, (d) two-spotted spider mite, (e) septoria leaf spot, (f) leaf mold, (g) late blight, (h) early blight and (i) bacterial spot.

Crops require continuous monitoring for early disease detection and thus the ability to apply proper mechanisms to prevent its spread and the loss of production [8].

The traditional methods used for the detection of plant diseases focus on the visual estimation of the disease by experts; studies of morphological characteristics to identify the pathogens; and molecular, serological, and microbiological diagnostic techniques [9]. The visual estimation method for plant disease identification is based on the analysis of characteristic disease symptoms (such as lesions, blight, galls and tumors) or visible signs of a pathogen (uredinospores of Pucciniales, mycelium or conidia of Erysiphales). Visual estimation is very subjective, as it is performed according to the experience of experts, so the accuracy of identification cannot be measured, and it is affected by temporal variation [10]. Microscopic methods focus on pathogen morphology for disease detection. However, these methods are expensive, time-consuming in the detection process and lead to low detection efficiency and poor reliability. In addition, farmers do not have the necessary knowledge to carry out the detection process, and agricultural experts cannot be in the field all the time to carry out proper monitoring.

New innovative techniques need to address the challenges and trends demanded by the new vision of agricultural production that requires higher accuracy levels and near real-time detection.

In recent years, different technologies such as image processing [11,12], pattern recognition [13,14] and computer vision [15,16] have rapidly developed and been applied to agriculture, specifically on automation of disease and pest detection processes. Traditional computer vision models face serious problems due to their complex preprocessing and design of image features that are time-consuming and labor-intensive. In addition, their

efficiency is conditioned by the accuracy in the design of feature extraction mechanisms and the learning algorithm [17].

Recently, the problem of plant disease detection has been addressed by deep learning technology, a subset of machine learning that is gaining momentum in disease identification due to the increase in computing power, storage capabilities and the availability of large data sets. Within the deep learning environment, one of the most widely used techniques for image classification, object detection and semantic segmentation are Convolutional Neural Networks (CNN) [18,19]. CNNs are useful for locating patterns in images, objects, and scenes by learning from the data obtained from the image for classification, eliminating the need for manual extraction of the features being searched for. CNN consist of several layers (such as convolutional, pooling and fully connected layers) to learn features from different training data [20,21]. This paper presents an architecture based on CNNs and data augmentation for early disease identification and classification in tomato leaves. The objective of the work is to implement a robust architecture that allows examining the relationship between the images of tomato leaves and the detection of a possible disease and performing a classification task to predict the type of disease with high accuracy levels.

The remainder of this article is organized as follows. Section 2 presents a brief discussion of previous research that has been conducted addressing the problem of disease identification in tomato. Section 3 explains in detail the CNN architecture proposed for tomato leaf disease identification and classification. A discussion of the experimental results obtained is presented in Section 4. Finally, Section 5 closes the paper with conclusions and future direction of the research work.

## 2. Related works

Plants disease detection has been studied for a long time. With respect to disease identification in tomatoes, much effort has been made using different tools such as classifiers focused on color [22,23], texture [24,25] or shape of tomato leaves [26]. Early efforts focused on support vector machines [27–30], decision trees [31,32] or neural network-based [33–35] classifiers. Visual spectrum images obtained from commercial cameras have been used for disease detection in tomato. The images obtained were processed under laboratory conditions, applying mechanisms such as stepwise multiple linear regression [36] and clustering process [37]. It is worth mentioning that the sample population for both works ranged between 22 and 47 for the first method and included 180 samples for the second experiment.

CNNs have rapidly become one of the preferred methods for disease detection in plants [38–40]. Some works have focused their efforts on identifying features with better quality through the process of eliminating the limitations generated by lighting conditions and uniformity in complex environment situations [41,42]. Some authors have developed real-time models to accelerate the process of disease detection in plants [43,44]. Other authors have created models that contribute to the early detection of plant diseases [45,46]. In [47], the authors make use of images of tomato leaves to discover different types of diseases. The authors apply artificial intelligence algorithms and CNN to perform a classification model to detect five types of diseases obtaining an accuracy of 96.55%. Some works evaluated the performance of deep neural network models applied to tomato leaf disease detection such as in [48], where the authors evaluated the LeNet, VGG16, ResNet and Xception models for the classification of nine types of diseases, determining that the VGG16 model is the one that obtained the best performance with an accuracy of 99.25%. In [49], the authors applied the AlexNet, GoogleNet and LeNet models to solve the same problem, obtaining accuracy results ranging between 94% and 95%. Agarwal et al. [50] developed their own CNN model based on the structure of VGG16 and compared it with different machine learning models (including random forest and decision trees) and deep learning models (VGG16, Inceptionv3 and MobileNet) to perform the classification of the 10 classes, obtaining an accuracy of 98.4%.

Several researches have focused on combining deep learning algorithms with machine learning algorithms to address and improve the accuracy of the classification problem,

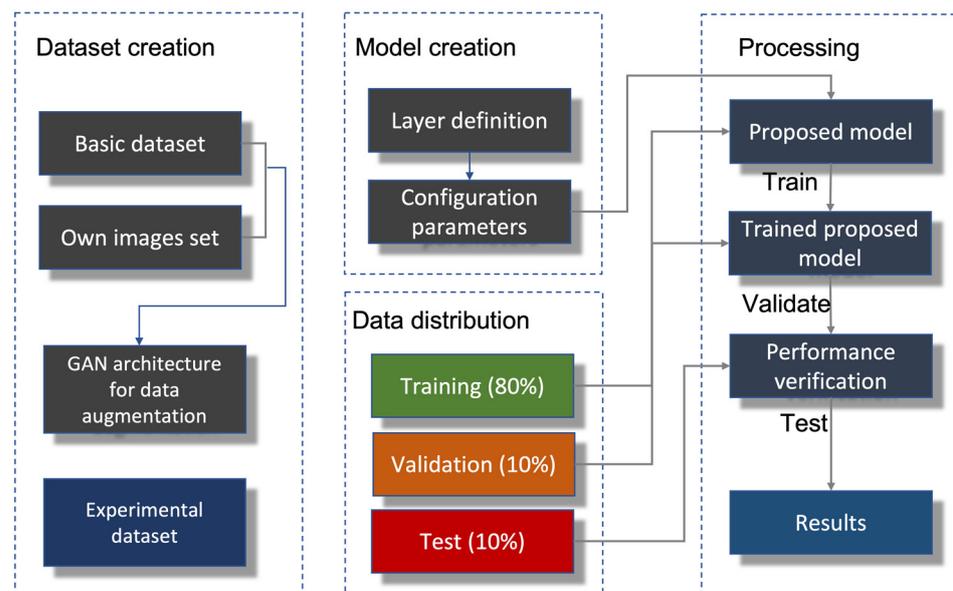
for example, MobileNetv2 and NASNetMobile that were used to extract features from leaves and those features were combined with classification networks such as random forest, support vector machines and multinomial logistic regression [51]. Other works have applied algorithms such as YOLOv3 [45], Faster R-CNN [52,53] and Mask R-CNN [54,55] to detect disease states in plants.

Some efforts have been made to reduce the computational cost and model size such as Gabor filters [56] and K-nearest neighbors (KNN) [57] that have been implemented to reduce computational costs and overhead generated by deep learning. In [58], the authors reduced the computational cost by using the SqueezeNet architecture and minimizing the number of  $3 \times 3$  filters.

### 3. Materials and Methods

In this section, we explain in detail the proposed architecture for the detection of diseases in tomato leaves. In general, the proposed architecture takes tomato leaves as input images and the output is a set of labels indicating (1) the type of disease in the image being analyzed or whether the leaf is healthy, (2) the label showing the predicted value obtained by our model, and (3) the prediction percentage.

Figure 2 shows the complete process of the algorithm that we applied for the process of detection and classification of diseases in tomato leaves. The global algorithm is composed of four stages: (a) creation of the experimental dataset, (b) creation of the proposed architecture, (c) distribution of the dataset, and (d) process of training and evaluation of the model.



**Figure 2.** Representation of the proposed architecture for tomato disease detection.

#### 3.1. Dataset Creation

As a first step, we proceeded to create the experimental dataset that would be used for training, validation, and performance evaluation of the proposed architecture. The public dataset available in [59] consists of 11,000 images that were the basis of our dataset. The images represent 10 categories, including nine types of diseases (tomato mosaic virus, target spot, bacterial spot, tomato yellow leaf curl virus, late blight, leaf mold, early blight, two-spotted spider mites, septoria leaf spot) and one category of healthy leaves. The dataset was complemented with 2500 images obtained from different crop fields in Mexico. The total number of images that made up our dataset was 13,500.

One of the problems that datasets face with deep neural network models is that when training the model, overfitting can occur, i.e., a model with high capacity may be able to “memorize” the dataset [60]. A technique known as data augmentation is used to avoid the

problem of overfitting. The goal of applying data augmentation is to increase the size of the dataset, and it is widely used in all fields [61]. Commonly, data augmentation is performed by two methods. The first method, known as the traditional method, aims to obtain a new image, which contains the same semantic information but does not have the ability of generalization. These methods include translation, rotation, flip, brightness adjustment, affine transformation, Gaussian noise, etc. The main drawbacks of these methods may be their poor quality and inadequate diversity.

Another method is the use of Generative Adversarial Networks (GANs), which are an approach to generative modeling using deep learning methods, such as CNNs, that aim to generate synthetic samples with the same characteristics as the given training distribution [62]. GAN models mainly consist of two parts, namely the generator and the discriminator [63]. The generator is a model used to generate new plausible examples from the problem domain. The discriminator is a model used to classify examples as real (from the domain) or fake (generated).

To create our experimental dataset, we made use of GAN to avoid the overfitting problem. To build our GAN, we define two separate networks: the generator network and the discriminator network. The first network receives a random noise, and from that number, the network generates images. The second network, the discriminator, defines whether the image it receives as input is “real” or not.

Because the images that complemented the dataset were not balanced for each category, the GAN network generated images that contributed to balance the dataset. The dataset was increased from 13,500 to 15,000 images, distributing the generated images in the different categories to create a balanced dataset.

### 3.2. Model Creation

Figure 3 shows the proposed CNN architecture for disease detection in tomato. The network has  $112 \times 112$  color images as input, which are normalized to  $(0, 1)$  values. The proposed convolutional network has four convolutional layers that use filters whose values were 16, 32, 64, and 128, respectively. These values were assigned in that order since the layers closer to the beginning of the model learn convolutional filters less effectively than the layers closer to the result. In addition, the kernel size, which represents the width and height of the 2D convolution window, was set to a value of  $3 \times 3$ . This value was the recommended value for the number of filters to be used. Finally, rectified linear unit (ReLU) was used as the activation model for each convolved node.

After applying the convolutional layer, the maximum clustering layer was applied to down-sample the acquired feature map and condense the most relevant features into patches. This process is repeated for each of the convolutional layers defined in the architecture.

The result of the last MaxPooling layer is passed to a MaxAveragePooling layer to be converted to a column vector and connected to the dense layer of 10 output nodes (which represent the 10 categories) used as softmax activation. Each node represents the probability of each category for the evaluated image. Table 1 shows the information of the layer structure of the proposed model.

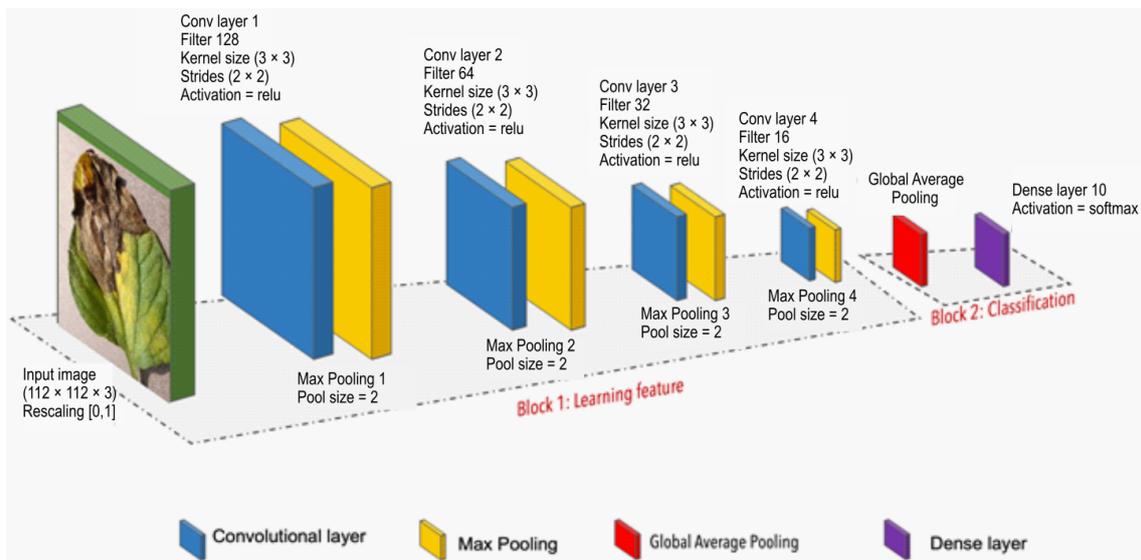


Figure 3. Representation of the proposed algorithm for tomato disease detection.

Table 1. Information on the layers structure of the proposed model.

Layers	Parameters
Conv2D	Filters: 128, kernel size: (3,3), activation: “relu”, input shape: (112,112,3)
MaxPool2D	Pool size: (2,2)
Conv2D	Filters: 64, kernel size: (3,3), activation: “relu”
MaxPool2D	Pool size: (2,2)
Conv2D	Filters: 32, kernel size: (3,3), activation: “relu”
MaxPool2D	Pool size: (2,2)
Conv2D	Filters: 16, kernel size: (3,3), activation: “relu”
MaxPool2D	Pool size: (2,2)
Dropout	Rate: 0.2
GlobalAveragePooling2D	
Dense	Units: 10, activation: “softmax”

### 3.3. Data Distribution

One of the most common strategies to split the dataset into training and validation sets is assigning percentages, for example, 70:30 or 80:20. However, one of the problems that can arise with this strategy is that it is uncertain whether high validation accuracy indicates a good model. When performing this division, it could happen that some information is missing in the data that are not used for training, causing a bias in the results.

We apply a k-fold cross-validation method to evaluate the performance of the model. The k-folds method tries to ensure that all features of the dataset are in the training and validation phases. The k-fold cross-validation method divides the dataset into subsets as k number. Therefore, it repeats the cross-validation method k times. Common values in machine learning are k = 3, k = 5, and k = 10. We use k = 5 to provide good trade-off of low computational cost and low bias in an estimate of model performance.

### 3.4. Model Creation

For the training process, we use Adam as the optimization algorithm. Adam updates network weights iterative based on training data. The loss function was categorical\_crossentropy, one of the most used loss functions for multi-class classification models

where there are two or more output labels. The number of epochs for the training and validation process was 200. The steps\_per\_epoch parameter was 12,000, and for the validation the parameter it was 3000. Table 2 shows a summary of some of the parameters used for the training and validation phase.

**Table 2.** Training Parameters for the Proposed Model.

Parameter	Value
Optimization algorithm	Adam
Loss function	Categorical cross entropy
Batch size	32
Number of epochs	200
Steps per epoch	12,000
Validation steps	3000
Activation function for conv layer	ReLU

## 4. Results

In this section, we describe the scenario setup and the results obtained in the performance evaluation process of the proposed model.

### 4.1. Environmental Setup

Our model was developed in Google Collaboratory, a free Python development environment that runs in the cloud. Google Collaboratory is widely used for the development of machine learning and deep learning projects. In our project, we use the following libraries: Tensorflow, an open-source library used for numerical computation and automated learning; Keras, a library used for the creation of neural networks; numpy, used for data analysis and mathematical calculations; matplotlib used for graph management and TensorBoard to visually inspect the different runs and graphs.

The model was trained with 200 epochs. We applied early stopping to monitor the performance of the model for the 200 epochs on a held-out validation set during the training to reduce overfitting and to improve the generalization of the neural network. For the evaluation of the model, the validation accuracy scheme allowed early stopping to be activated during the process.

Since our problem is a multi-class classification model, we use the Adam algorithm as the optimizing algorithm. In addition, the cross-entropy categorical loss function was used due to the nature of the multi-class classification environment. During the training process, we implemented checkpoints to save the model with the best validation accuracy, and thus be able to load it later to continue training from the saved state if necessary.

### 4.2. Evaluation Metrics

To analyze the performance of our model, the following four metrics were considered. The first metric to evaluate was accuracy, which represents the behavior of the model across all classes. Accuracy is calculated as the ratio between the number of correct predictions to the total number of predictions (Equation (1)).

Precision was our second metric, which represents the accuracy of the model in classifying a sample as positive. This parameter is calculated as the ratio of the number of positive samples correctly classified to the total number of samples classified as positive (Equation (2)).

We also analyzed the recall parameter, which measures the ability of the model to detect positive samples and is calculated as the ratio of the number of positive samples correctly classified to the total number of positive samples (Equation (3)).

Finally, we analyzed the *F1* score parameter. This metric combines the precision and recall measures to obtain a single value. This value is calculated by taking the harmonic mean between precision and recall (Equation (4)).

The following equations were used to calculate accuracy, precision, recall and F1 score:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 \text{ Score} = 2 \times \frac{(precision \times recall)}{(precision + recall)} \quad (4)$$

### 4.3. Results and Discussion

In this section, we analyze the results obtained in the evaluation of the performance of the proposed CNN model in tomato crops. We compare our results with some of the proposed models published in the literature.

#### 4.3.1. Validation of the Proposed Model

The validation of the model was analyzed by applying the k-fold cross-validation procedure to estimate the performance of our algorithm on the tomato images dataset. We define a k-value of 5 to split the dataset. The Scikit-Learn machine learning library was used to implement the k-fold method returning a list of scores calculated for each of our five folds.

Figure 4 shows the results obtained by applying the k-folds method to evaluate the performance of the model. The results show a stability of the model, as it is observed in the different metrics analyzed. There is a very similar behavior with the five folds both in the training phase and in the validation phase, demonstrating that there is no overfitting in the proposed model.



Figure 4. Results obtained for k-folds.

Figure 5 demonstrates the performance of our model in the training and validation stages for identification and classification of tomato leaf diseases. The results achieved a training accuracy of 99.99%. The time used for the training process was 6234 s in the MGPU (Multiple-Graphics Processing Unit) environment. The proposed model achieved a validation accuracy of 99.64% in leaf disease classification.

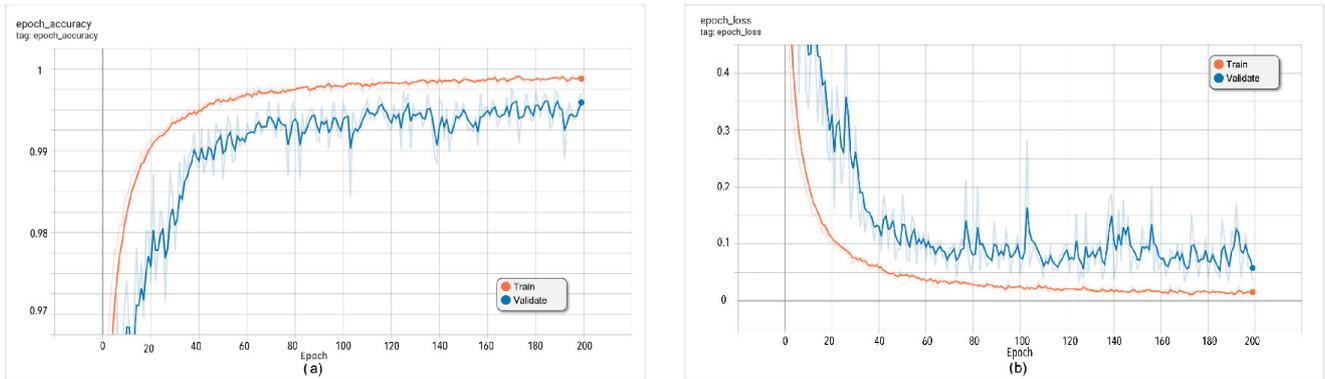


Figure 5. Results obtained of the proposed model during the training and validation phases: (a) accuracy and (b) loss.

Figure 6 shows the confusion matrix obtained in the evaluation of the proposed model. The confusion matrix shows the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) values obtained for each class evaluated [64].

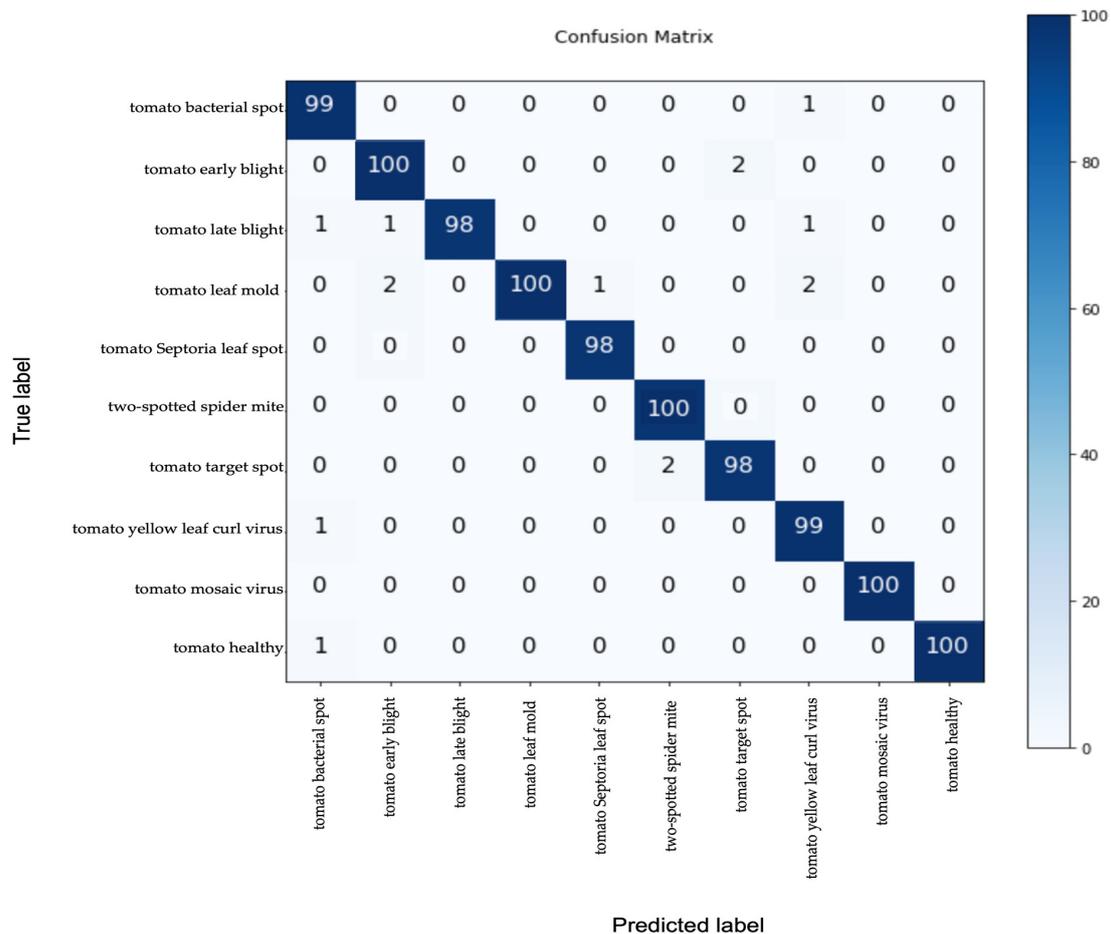


Figure 6. Confusion matrix of the proposed model.

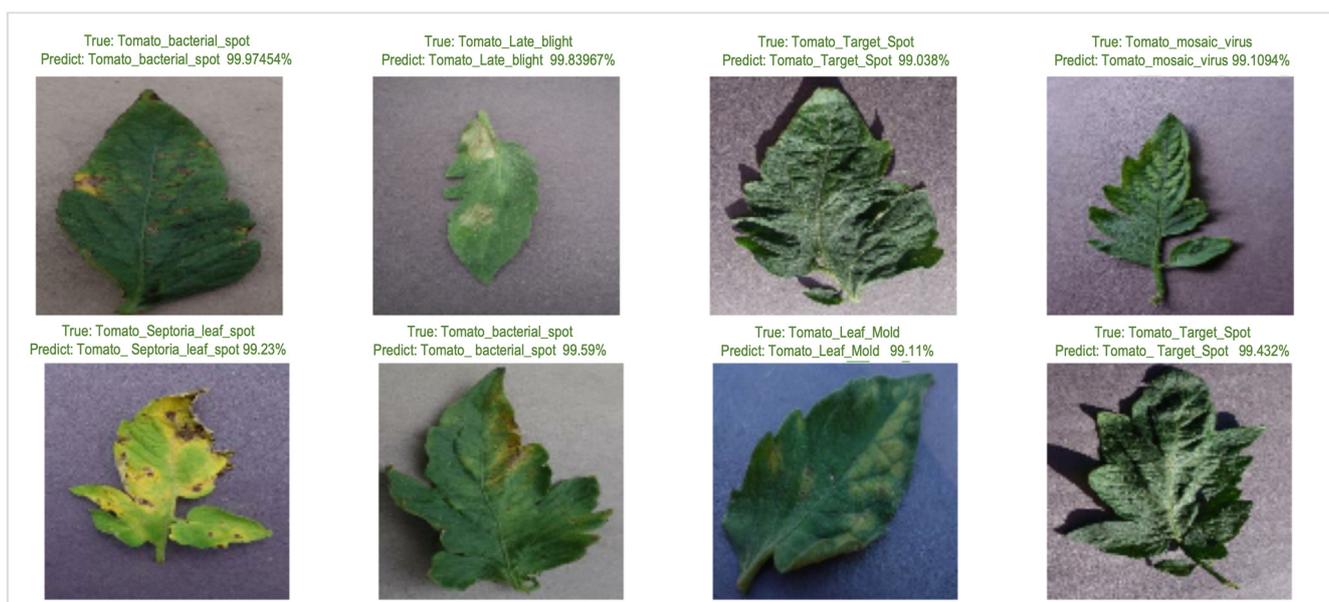
According to the results, which are reflected in the confusion matrix, we can see that the proposed model was able to predict half of the classes that were evaluated using the test dataset with a 100% accuracy. For the rest of the classes, the model reached an accuracy level of at least 98%, thus obtaining better values than those of several of the works proposed in the literature.

Table 3 presents the results obtained in the classification performance of the proposed model on each of the classes defined within the experimental dataset. According to the data reflected in the table, the value obtained for the recall metric is high for each category defined in the dataset; this allows inferring the performance of the proposed model, which is able to correctly classify the corresponding disease with accuracy higher than 98%.

**Table 3.** Class-wise Performance of the Proposed Model.

Class	Precision	Recall	F1 Score	Support
Tomato bacterial spot	0.99	0.990	0.99	100
Tomato early blight	1	1	0.98	100
Tomato late blight	0.97	0.98	0.97	100
Tomato leaf mold	1	1	0.99	100
Tomato Septoria leaf spot	0.99	0.98	0.97	100
Tomato Two-spotted spider mite	1	1	0.98	100
Tomato target spot	0.99	0.98	0.99	100
Tomato yellow leaf curl virus	0.98	0.98	0.98	100
Tomato mosaic virus	1	1	0.99	100
Tomato healthy	1	1	0.99	100

The architecture and weights obtained from the proposed model were saved as a hierarchical data file to be used during the prediction process. The prediction process uses a dataset with a total of 1350 images. The matplotlib library was used to visualize the prediction result. For each prediction, the image, the true result, and the result of the prediction made with the proposed model were displayed, together with the percentage of accuracy. Figure 7 shows some results of the predictions made by the model.



**Figure 7.** Sample predicted images using the proposed model.

#### 4.3.2. Comparison of the Model

Finally, our model was compared with other techniques proposed in the literature (Widiyanto et al. [65], Afif Al Mamun et al. [66], Kaur et al. [67], AlexNet [68]; Inception-v3-

Net, ResNet-50 and VGG16Net [69]). Figure 8 presents the results of the comparison and shows that for the accuracy and recall metrics, the proposed model obtained the best results, reaching an accuracy of 99.9%. With respect to the precision metric, the proposed algorithm had a result only lower than the VGG16Net technique, but with a result of 0.99. For the F1 metric, the proposed model had a similar result to that of the VGG16Net technique.

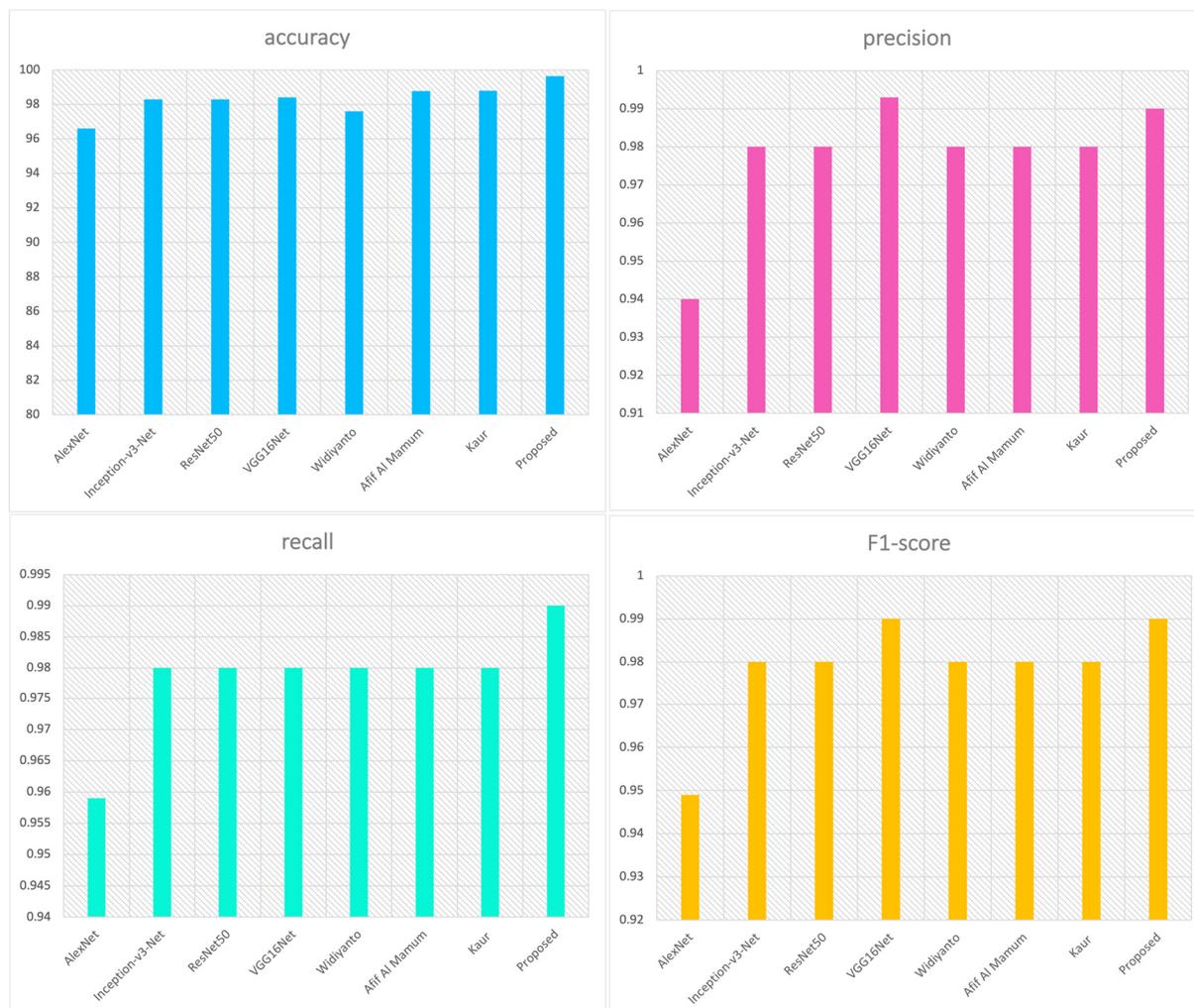


Figure 8. Performance comparison of proposed model and existing models.

In addition, a comparison was made of the complexity of the proposed model and some of the other models included in the comparison (data were not obtained for some of the models used in the comparison). Specifically, the number of trainable parameters and the size of the model were analyzed. The data obtained are shown in Table 4. Finally, Table 5 shows a summary of the performance of the models using the metrics accuracy, precision, recall and F1 score.

Table 4. Complexity comparison of proposed model and existing models.

	ResNet	VGG16Net	Inception-v3-Net	AlexNet	Proposed
Trainable parameters (Millions)	26.7	39.4	24.9	44.7	5.6
Model size (MB)	98	128	92	133	36

**Table 5.** Class-wise Performance of the Proposed Model.

Reference	Accuracy	Precision	Recall	F1 Score
Widiyanto, et al. (2019)	97.6	0.98	0.98	0.98
Afif Al Mamum et al. (2020)	98.77	0.98	0.98	0.98
Kaur et al. (2019)	98.8	0.98	0.98	0.98
Proposed model	99.64	0.99	0.99	0.99

## 5. Conclusions

In this research, we propose an architecture based on CNNs to identify and classify nine different types of tomato leaf diseases. The complexity in detecting the type of disease lies in the fact that the leaves deteriorate in a similar way in most of the tomato diseases. It means that it is necessary to develop a deep image analysis to judge the types of tomato leaf diseases with a proper accuracy level.

The CNN that we design is a high-performance deep learning network that allows us to have a complex image processing and feature extraction through four modules: the module dataset creation that makes an experimental dataset using public datasets and photographs taken in the fields of the country; model creation that is in charge of parameters configuration and layers definition; data distribution to train, validate and test data; and processing for the optimization and performance verification.

We evaluate the performance of our model via accuracy, precision, recall and the F1-score metrics. The results showed a training accuracy of 99.99% and a validation accuracy of 99.64% in the leaf disease classification. The model correctly classifies the corresponding disease with a precision of 0.99 and an F1 score of 0.99. The recall metric has a value of 0.99 on the classification of the nine tomato diseases that we analyzed.

The resulting confusion matrix describes that our classification model was able to predict half of the classes that were evaluated using the test dataset with a 100% accuracy. For the rest of the classes, the model reached an accuracy level of 98%, thus obtaining better values than those of several of the works proposed in the literature.

**Author Contributions:** Conceptualization, A.G.-I.; Methodology, A.G.-I. and A.R.-M.; Software A.G.-I.; Validation, A.G.-I. and A.R.-M.; Formal analysis, A.G.-I.; Resources, A.R.-M.; Data curation, A.G.-I.; Writing—review & editing, A.G.-I. and A.R.-M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially funded by the State Research Agency of Spain under grant number PID2020-116377RB-C21.

**Data Availability Statement:** The datasets generated during the current study are available from authors on reasonable request.

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

## References

1. Food and Agriculture Organization of the United Nations. “FAOSTAT” Crops and Livestock Products. Available online: <https://www.fao.org/faostat/en/#data/QCL> (accessed on 19 October 2022).
2. Los Productos Agropecuarios Más Exportados. 1 July 2022. Available online: <https://mundi.io/exportacion/exportacion-productos-agropecuarios-mexico/> (accessed on 2 November 2022).
3. Ritchie, H.; Rosado, P.; Roser, M. Agricultural Production—Crop Production Across the World. 2020. Available online: <https://ourworldindata.org/agricultural-production> (accessed on 2 December 2022).
4. Food and Agriculture Organization of the United Nations. FAO—News Article: Climate Change Fans Spread of Pests and Threatens Plants and Crops—New FAO Study. Available online: <https://www.fao.org/news/story/en/item/1402920/icode/> (accessed on 19 October 2022).
5. Gobalakrishnan, N.; Pradeep, K.; Raman, C.J.; Ali, L.J.; Gopinath, M.P. A Systematic Review on Image Processing and Machine Learning Techniques for Detecting Plant Diseases. In Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 28–30 July 2020; pp. 0465–0468. [CrossRef]

6. Damicone, J.; Brandenberger, L. Common Diseases of Tomatoes: Part I. Diseases Caused by Fungi—Oklahoma State University. 2016. Available online: <https://extension.okstate.edu/fact-sheets/common-diseases-of-tomatoes-part-i-diseases-caused-by-fungi.html> (accessed on 19 October 2022).
7. Ahmad, A.; Saraswat, D.; El Gamal, A. A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. *Smart Agric. Technol.* **2023**, *3*, 100083. [[CrossRef](#)]
8. DeChant, C.; Wiesner-Hanks, T.; Chen, S.; Stewart, E.L.; Yosinski, J.; Gore, M.A.; Nelson, R.J.; Lipson, H. Automated Identification of Northern Leaf Blight-Infected Maize Plants from Field Imagery Using Deep Learning. *Phytopathology* **2017**, *107*, 1426–1432. [[CrossRef](#)]
9. Bock, C.H.; Poole, G.H.; Parker, P.E.; Gottwald, T.R. Plant Disease Severity Estimated Visually, by Digital Photography and Image Analysis, and by Hyperspectral Imaging. *Crit. Rev. Plant Sci.* **2010**, *29*, 59–107. [[CrossRef](#)]
10. Bock, C.H.; Parker, P.E.; Cook, A.Z.; Gottwald, T.R. Visual Rating and the Use of Image Analysis for Assessing Different Symptoms of Citrus Canker on Grapefruit Leaves. *Plant Dis.* **2008**, *92*, 530–541. [[CrossRef](#)] [[PubMed](#)]
11. Devaraj, A.; Rathan, K.; Jaahnavi, S.; Indira, K. Identification of Plant Disease using Image Processing Technique. In Proceedings of the 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 4–6 April 2019; pp. 0749–0753. [[CrossRef](#)]
12. Mugithe, P.K.; Mudunuri, R.V.; Rajasekar, B.; Karthikeyan, S. Image Processing Technique for Automatic Detection of Plant Diseases and Alerting System in Agricultural Farms. In Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 28–30 July 2020; pp. 1603–1607. [[CrossRef](#)]
13. Phadikar, S.; Sil, J. Rice disease identification using pattern recognition techniques. In Proceedings of the 2008 11th International Conference on Computer and Information Technology, Khulna, Bangladesh, 24–27 December 2008; pp. 420–423. [[CrossRef](#)]
14. Sarayloo, Z.; Asemani, D. Designing a classifier for automatic detection of fungal diseases in wheat plant: By pattern recognition techniques. In Proceedings of the 2015 23rd Iranian Conference on Electrical Engineering, Tehran, Iran, 10–14 May 2015; pp. 1193–1197. [[CrossRef](#)]
15. Thangadurai, K.; Padmavathi, K. Computer Vision image Enhancement for Plant Leaves Disease Detection. In Proceedings of the 2014 World Congress on Computing and Communication Technologies, Trichirappalli, India, 27 February–1 March 2014; pp. 173–175. [[CrossRef](#)]
16. Yong, Z.; Tonghui, R.; Changming, L.; Chao, W.; Jiya, T. Research on Recognition Method of Common Corn Diseases Based on Computer Vision. In Proceedings of the 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 24–25 August 2019; Volume 1, pp. 328–331. [[CrossRef](#)]
17. Khirade, S.D.; Patil, A.B. Plant Disease Detection Using Image Processing. In Proceedings of the 2015 International Conference on Computing Communication Control and Automation, Pune, India, 26–27 February 2015; pp. 768–771. [[CrossRef](#)]
18. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [[CrossRef](#)]
19. Li, L.; Zhang, S.; Wang, B. Plant Disease Detection and Classification by Deep Learning—A Review. *IEEE Access* **2021**, *9*, 56683–56698. [[CrossRef](#)]
20. Lee, S.H.; Chan, C.S.; Wilkin, P.; Remagnino, P. Deep-plant: Plant identification with convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada; 2015; pp. 452–456. [[CrossRef](#)]
21. Zhang, Y.; Song, C.; Zhang, D. Deep Learning-Based Object Detection Improvement for Tomato Disease. *IEEE Access* **2020**, *8*, 56607–56614. [[CrossRef](#)]
22. Widiyanto, S.; Wardani, D.T.; Pranata, S.W. Image-Based Tomato Maturity Classification and Detection Using Faster R-CNN Method. In Proceedings of the 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSTIT), Ankara, Turkey, 21–23 October 2021; pp. 130–134. [[CrossRef](#)]
23. Zhou, X.; Wang, P.; Dai, G.; Yan, J.; Yang, Z. Tomato Fruit Maturity Detection Method Based on YOLOV4 and Statistical Color Model. In Proceedings of the 2021 IEEE 11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Jiaxing, China, 27–31 July 2021; pp. 904–908. [[CrossRef](#)]
24. Hlaing, C.S.; Zaw, S.M.M. Tomato Plant Diseases Classification Using Statistical Texture Feature and Color Feature. In Proceedings of the 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), Singapore, 6–8 June 2018; pp. 439–444. [[CrossRef](#)]
25. Lu, J.; Shao, G.; Gao, Y.; Zhang, K.; Wei, Q.; Cheng, J. Effects of water deficit combined with soil texture, soil bulk density and tomato variety on tomato fruit quality: A meta-analysis. *Agric. Water Manag.* **2021**, *243*, 106427. [[CrossRef](#)]
26. Kaur, S.; Pandey, S.; Goel, S. Plants Disease Identification and Classification Through Leaf Images: A Survey. *Arch. Comput. Methods Eng.* **2018**, *26*, 507–530. [[CrossRef](#)]
27. Bhagat, M.; Kumar, D.; Haque, I.; Munda, H.S.; Bhagat, R. Plant Leaf Disease Classification Using Grid Search Based SVM. In Proceedings of the 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 28–29 February 2020; pp. 1–6. [[CrossRef](#)]
28. Rani, F.A.P.; Kumar, S.N.; Fred, A.L.; Dyson, C.; Suresh, V.; Jeba, P.S. K-means Clustering and SVM for Plant Leaf Disease Detection and Classification. In Proceedings of the 2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC), Nagercoil, India, 7–8 March 2019; pp. 1–4. [[CrossRef](#)]

29. Padol, P.B.; Yadav, A.A. SVM classifier based grape leaf disease detection. In Proceedings of the 2016 Conference on Advances in Signal Processing (CASP), Pune, India, 9–11 June 2016; pp. 175–179. [[CrossRef](#)]
30. Mokhtar, U.; Ali, M.A.S.; Hassenian, A.E.; Hefny, H. Tomato leaves diseases detection approach based on Support Vector Machines. In Proceedings of the 2015 11th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 29–30 December 2015; pp. 246–250. [[CrossRef](#)]
31. Sabrol, H.; Satish, K. Tomato plant disease classification in digital images using classification tree. In Proceedings of the 2016 International Conference on Communication and Signal Processing (ICCSF), Melmaruvathur, India, 6–8 April 2016; pp. 1242–1246. [[CrossRef](#)]
32. Chopda, J.; Raveshiya, H.; Nakum, S.; Nakrani, V. Cotton Crop Disease Detection using Decision Tree Classifier. In Proceedings of the 2018 International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, India, 5 January 2018; pp. 1–5. [[CrossRef](#)]
33. Molina, F.; Gil, R.; Bojacá, C.; Gómez, F.; Franco, H. Automatic detection of early blight infection on tomato crops using a color based classification strategy. In Proceedings of the 2014 XIX Symposium on Image, Signal Processing and Artificial Vision, Armenia, Colombia, 17–19 September 2014; pp. 1–5. [[CrossRef](#)]
34. Pratheba, R.; Sivasangari, A.; Saraswady, D. Performance analysis of pest detection for agricultural field using clustering techniques. In Proceedings of the 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014], Nagercoil, India, 20–21 March 2014; pp. 1426–1431. [[CrossRef](#)]
35. Shijie, J.; Peiyi, J.; Siping, H.; Haibo, L. Automatic detection of tomato diseases and pests based on leaf images. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 2510–2537. [[CrossRef](#)]
36. Jones, C.; Jones, J.; Lee, W.S. Diagnosis of bacterial spot of tomato using spectral signatures. *Comput. Electron. Agric.* **2010**, *74*, 329–335. [[CrossRef](#)]
37. Borges, D.L.; Guedes, S.T.D.M.; Nascimento, A.R.; Melo-Pinto, P. Detecting and grading severity of bacterial spot caused by *Xanthomonas* spp. in tomato (*Solanum lycopersicon*) fields using visible spectrum images. *Comput. Electron. Agric.* **2016**, *125*, 149–159. [[CrossRef](#)]
38. Lakshmanarao, A.; Babu, M.R.; Kiran, T.S.R. Plant Disease Prediction and classification using Deep Learning ConvNets. In Proceedings of the 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV), Gandhinagar, India, 24–26 September 2021; pp. 1–6. [[CrossRef](#)]
39. Militante, S.V.; Gerardo, B.D.; Dionisio, N.V. Plant Leaf Detection and Disease Recognition using Deep Learning. In Proceedings of the 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 3–6 October 2019; pp. 579–582. [[CrossRef](#)]
40. Marzougui, F.; Elleuch, M.; Kherallah, M. A Deep CNN Approach for Plant Disease Detection. In Proceedings of the 2020 21st International Arab Conference on Information Technology (ACIT), Giza, Egypt, 28–30 November 2020; pp. 1–6. [[CrossRef](#)]
41. Ngugi, L.C.; Abdelwahab, M.; Abo-Zahhad, M. Tomato leaf segmentation algorithms for mobile phone applications using deep learning. *Comput. Electron. Agric.* **2020**, *178*, 105788. [[CrossRef](#)]
42. Elhassouny, A.; Smarandache, F. Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks. In Proceedings of the 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), Agadir, Morocco, 22–24 July 2019; pp. 1–4. [[CrossRef](#)]
43. Mattihalli, C.; Gedefaye, E.; Endalamaw, F.; Necho, A. Real Time Automation of Agriculture Land, by automatically Detecting Plant Leaf Diseases and Auto Medicine. In Proceedings of the 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), Krakow, Poland, 16–18 May 2018; pp. 325–330. [[CrossRef](#)]
44. Divyashri, P.; Pinto, L.A.; Mary, L.; Manasa, P.; Dass, S. The Real-Time Mobile Application for Identification of Diseases in Coffee Leaves using the CNN Model. In Proceedings of the 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 4–6 August 2021; pp. 1694–1700. [[CrossRef](#)]
45. Liu, J.; Wang, X. Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model. *Plant Methods* **2020**, *16*, 83. [[CrossRef](#)] [[PubMed](#)]
46. Khasawneh, N.; Faouri, E.; Fraiwan, M. Automatic Detection of Tomato Diseases Using Deep Transfer Learning. *Appl. Sci.* **2022**, *12*, 8467. [[CrossRef](#)]
47. Mim, T.T.; Sheikh, M.H.; Shampa, R.A.; Reza, M.S.; Islam, M.S. Leaves Diseases Detection of Tomato Using Image Processing. In Proceedings of the 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART), Moradabad, India, 22–23 November 2019; pp. 244–249. [[CrossRef](#)]
48. Kumar, A.; Vani, M. Image Based Tomato Leaf Disease Detection. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–6. [[CrossRef](#)]
49. Tm, P.; Pranathi, A.; SaiAshritha, K.; Chittaragi, N.B.; Koolagudi, S.G. Tomato Leaf Disease Detection Using Convolutional Neural Networks. In Proceedings of the 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, India, 2–4 August 2018; pp. 1–5. [[CrossRef](#)]
50. Agarwal, M.; Gupta, S.K.; Biswas, K. Development of Efficient CNN model for Tomato crop disease identification. *Sustain. Comput. Inform. Syst.* **2020**, *28*, 100407. [[CrossRef](#)]
51. Al-Gaashani, M.S.A.M.; Shang, F.; Muthanna, M.S.A.; Khayyat, M.; El-Latif, A.A.A. Tomato leaf disease classification by exploiting transfer learning and feature concatenation. *IET Image Process.* **2022**, *16*, 913–925. [[CrossRef](#)]

52. Pathan, S.M.K.; Ali, M.F. Implementation of Faster R-CNN in Paddy Plant Disease Recognition System. In Proceedings of the 2019 3rd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), Rajshahi, Bangladesh, 26–28 December 2019; pp. 189–192. [CrossRef]
53. Zhou, G.; Zhang, W.; Chen, A.; He, M.; Ma, X. Rapid Detection of Rice Disease Based on FCM-KM and Faster R-CNN Fusion. *IEEE Access* **2019**, *7*, 143190–143206. [CrossRef]
54. Mu, W.; Jia, Z.; Liu, Y.; Xu, W.; Liu, Y. Image Segmentation Model of Pear Leaf Diseases Based on Mask R-CNN. In Proceedings of the 2022 International Conference on Image Processing and Media Computing (ICIPMC), Xi'an, China, 27–29 May 2022; pp. 41–45. [CrossRef]
55. Wang, Q.; Qi, F.; Sun, M.; Qu, J.; Xue, J. Identification of Tomato Disease Types and Detection of Infected Areas Based on Deep Convolutional Neural Networks and Object Detection Techniques. *Comput. Intell. Neurosci.* **2019**, *2019*, 9142753. [CrossRef]
56. Kirange, D. Machine Learning Approach towards Tomato Leaf Disease Classification. *Int. J. Adv. Trends Comput. Sci. Eng.* **2020**, *9*, 490–495. [CrossRef]
57. Lu, J.; Ehsani, R.; Shi, Y.; De Castro, A.I.; Wang, S. Detection of multi-tomato leaf diseases (late blight, target and bacterial spots) in different stages by using a spectral-based sensor. *Sci. Rep.* **2018**, *8*, 2793. [CrossRef]
58. Durmuş, H.; Güneş, E.O.; Kırıcı, M. Disease detection on the leaves of the tomato plants by using deep learning. In Proceedings of the 2017 6th International Conference on Agro-Geoinformatics, Fairfax, VA, USA, 7–10 August 2017; pp. 1–5. [CrossRef]
59. Tomato Leaf Disease Detection. Available online: <https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf> (accessed on 24 October 2022).
60. Konidaris, F.; Tagaris, T.; Sdraka, M.; Stafylopatis, A. Generative Adversarial Networks as an Advanced Data Augmentation Technique for MRI Data. In Proceedings of the VISIGRAPP, Prague, Czech Republic, 25–27 February 2019.
61. Kukacka, J.; Golkov, V.; Cremers, D. Regularization for Deep Learning: A Taxonomy. *arXiv* **2017**, arXiv:1710.10686.
62. Pandian, J.A.; Kumar, V.D.; Geman, O.; Hnatiuc, M.; Arif, M.; Kanchanadevi, K. Plant Disease Detection Using Deep Convolutional Neural Network. *Appl. Sci.* **2022**, *12*, 6982. [CrossRef]
63. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
64. Geetharamani, G.; Arun Pandian, J. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput. Electr. Eng.* **2019**, *76*, 323–338. [CrossRef]
65. Widiyanto, S.; Fitrianto, R.; Wardani, D.T. Implementation of Convolutional Neural Network Method for Classification of Diseases in Tomato Leaves. In Proceedings of the 2019 Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia, 16–17 October 2019; pp. 1–5. [CrossRef]
66. Mamun, M.A.A.; Karim, D.Z.; Pinku, S.N.; Bushra, T.A. TLNet: A Deep CNN model for Prediction of tomato Leaf Diseases. In Proceedings of the 2020 23rd International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 19–21 December 2020; pp. 1–6. [CrossRef]
67. Kaur, M.; Bhatia, R. Development of an Improved Tomato Leaf Disease Detection and Classification Method. In Proceedings of the 2019 IEEE Conference on Information and Communication Technology, Allahabad, India, 6–8 December 2019; pp. 1–5. [CrossRef]
68. Nachtigall, L.; Araujo, R.; Nachtigall, G.R. Classification of apple tree disorders using convolutional neural networks. In Proceedings of the 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), San Jose, CA, USA, 6–8 November 2016; pp. 472–476.
69. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [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.