

## Article

# Detection and Classification of Tomato Crop Disease Using Convolutional Neural Network

Gnanavel Sakkarvarthi <sup>1</sup>, Godfrey Winster Sathianesan <sup>1,\*</sup>, Vetri Selvan Murugan <sup>2</sup>,  
Avulapalli Jayaram Reddy <sup>3</sup>, Prabhu Jayagopal <sup>3,\*</sup> and Mahmoud Elsisy <sup>4,5,\*</sup>

- <sup>1</sup> Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur 603203, India
- <sup>2</sup> Department of Artificial Intelligence and Data Science, Panimalar Engineering College, Chennai 600123, India
- <sup>3</sup> School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, India
- <sup>4</sup> Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo 11629, Egypt
- <sup>5</sup> Department of Electrical Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 807618, Taiwan
- \* Correspondence: godfreys@srmist.edu.in (G.W.S.); j.prabhu@vit.ac.in (P.J.); mahmoudelsisi@nkust.edu.tw (M.E.)

**Abstract:** Deep learning is a cutting-edge image processing method that is still relatively new but produces reliable results. Leaf disease detection and categorization employ a variety of deep learning approaches. Tomatoes are one of the most popular vegetables and can be found in every kitchen in various forms, no matter the cuisine. After potato and sweet potato, it is the third most widely produced crop. The second-largest tomato grower in the world is India. However, many diseases affect the quality and quantity of tomato crops. This article discusses a deep-learning-based strategy for crop disease detection. A Convolutional-Neural-Network-based technique is used for disease detection and classification. Inside the model, two convolutional and two pooling layers are used. The results of the experiments show that the proposed model outperformed pre-trained InceptionV3, ResNet 152, and VGG19. The CNN model achieved 98% training accuracy and 88.17% testing accuracy.

**Keywords:** Convolution Neural Network (CNN); Tomato Leaves Dataset (DLD); Data Augmentation (DA); Deep Learning (DL); Visual Geometry Group (VGG); Residual Network (ResNet)



**Citation:** Sakkarvarthi, G.; Sathianesan, G.W.; Murugan, V.S.; Reddy, A.J.; Jayagopal, P.; Elsisy, M. Detection and Classification of Tomato Crop Disease Using Convolutional Neural Network. *Electronics* **2022**, *11*, 3618. <https://doi.org/10.3390/electronics11213618>

Academic Editor: Stefanos Kollias

Received: 8 October 2022

Accepted: 3 November 2022

Published: 6 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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

In the world of agricultural need, the automatic detection of plant diseases using plant leaves is of major importance. Additionally, the early and timely detection of plant diseases improves agricultural production and quality [1]. Farmers in most agricultural countries lose significant amounts of money each year to crop diseases [2]. India is an agrarian nation, with agriculture accounting for a significant portion of the Gross Domestic Product (GDP). Agriculture accounts for 16% of India's Gross Domestic Product and 10% of total exports. Approximately 75% of India's population is dependent on agriculture, either directly or indirectly [3].

Tomatoes are essential due to their high market demand and nutritional value. These antioxidants are very important for our overall health. The production of this famous commodity can be hampered by insects and pests that attack tomato plants and cause various diseases. To treat diseases on tomato plants by hand, farmers need to know about the disease. Farmers face many problems every year when they try to grow healthy crops. Insects and other pests damage the production line and slow it down so that it cannot produce as much as it could. This is a big problem for our economy, and especially for our farmers.

Although farmers use a variety of pesticides and insecticides to protect tomato plants from disease, they often fail to have knowledge about the disease and how to avoid it. The excessive use of pesticides and insecticides is dangerous to human health and survival. Crop damage can be caused by misdiagnosing diseases and using too many or too few pesticides.

Diagnosing tomato plant diseases is critical for optimal production. In contrast, manually diagnosing tomato diseases by closely observing the plants is a time-consuming and difficult process. Farmers often find it difficult to consult experts in remote locations and take preventive measures against rare diseases. Lacking prior knowledge, the visual examination of plants may lead to incorrect disease prognosis. This also leads to the use of ineffective preventive measures. So, the use of a machine can find diseased tomato plants, figure out what kind of disease they are affected by, and use that information to help the rest of the crop grow more efficiently and with fewer losses.

## 2. Literature Survey

The Convolutional Neural Network (CNN) ResNet101 architecture was adapted to detect anthracnose diseases in olives using hyper spectral images. The collected olives which were free of external effect were inoculated with water and fungus. In order to obtain a dataset with different growth stages of flaws, images were captured on different days. Intending to achieve high accuracy, a data augmentation method was utilized. It increased the low number of captured samples [4].

Four types of cassava plant diseases, cassava mosaic disease, cassava brown streak, cassava green mite, and cassava bacterial blight, were detected and classified using the Convolutional Neural Network. The dataset had high class imbalance, was poorly contrasted, small in size, had a low resolution and skewed towards specific classes. This kind of dataset will affect the model accuracy. Therefore, to address the abovementioned issues, the authors used existing techniques, which were Contrast Limited Adaptive Histogram Equalization (CLAHE), focal length, class weight, SMOTE (Synthetic Minority Over Sampling Technique), and data augmentation. Then, they proved that after the pre-processing process, the accuracy of the cassava diseases detection model increased [5].

The CNN ResNet18 architecture was used to predict blackleg disease in potato plants. During image acquisition, LED strips were installed inside an enclosed camera box to avoid variance in the illumination condition. To block ambient light, imaging regions were attached with black rubber strips, and to obtain images in real-world positions, RTK-GNSS receivers were installed in an agriculture vehicle. A resizing process was performed to prevent image distortion. The deficiency in the first week after the emergence of plants taken for the selection of isolated plants should have solved the problem of plant overlap. Data augmentation was not used. Only the transfer learning strategy was applied [6].

A mobile pest monitoring system was developed using the RPSN (Rice Plant hopper Search Network) to detect rice plant hoppers using an image-segmentation-based CNN. The equipment used to collect the rice plant hopper dataset consisted of multiple sensors. Multiple high-quality affected regions were extracted from large-scale insect images. Additionally, the SSM (Sensitive Score Matrix) was applied to score the insect portions of the images. It enhanced the boundary box regression and the classification accuracy [7].

To identify and classify seventeen diseases in five different crops, the crop conditional model was developed, which integrated distinct CNN architecture with crop Meta information. These crops were barley, rice, wheat, corn, and rapeseed. The model easily learned similar disease symptoms in different crops from the obtained robust features of a large-size multi-crop dataset, which reduced the intricacy of the classification function [8].

A GPDCNN (Global Pooling Dilated Convolutional Neural Network) model was developed to identify cucumber leaf disease. The advantage of this model is that global pooling layers are combined with fully connected layers with the same discriminate formation. The increased receptive field does not lead to an increase in the computation complexity. The applied augmented CNN helps to obtain spatial resolution without in-

creasing the training parameters. The combined GPDCNN model offers the advantages of both global pooling and extended convolution [9].

A novel method based on DCNN was developed by Ashraf et al. to classify various medical images such as body organs. The last three layers of the GoogLeNet transfer learning method were replaced with a fully connected layer, a soft-max layer, and a classification output layer. The result of the fine-tuned approach exhibited the best result when classifying various medical images. Some researchers used a dataset with fewer images, which may have affected the training performance and increased the misclassification ratio [10].

The Deep Leaf Disease Prediction Framework (DLDPF) was developed to predict apple leaf diseases using a novel algorithm, which integrated a deep CNN and transfer learning with cascade inception. It consumed less memory and training time and increased the accuracy during the apple leaf diseases classification task compared to other models, including VggNet-16, GoogLeNet, and ResNet-20 [11].

The corn plant disease recognition model was implemented based on a DCNN and deployed in a USB-based neural computing stick made by Intel movidius, which was integrated with raspberry pi. Inside the DCNN, the pooling combination was adjusted, and hyper parameters were tuned. A pre-trained classifier has to have such a plethora of variables. It was optimized to obtain the inference of captured images in real time and achieved accuracy of 88.66%. The developed standalone device facilitates detection and addresses the diseases that affect corn plants without the use of the internet. The accuracy is much lower and needs to be improved. There is diversity in pooling operation, data augmentation and tuning, and optimization of hyper parameters [12].

A wrapper-based feature selection algorithm known as adaptive particle grey wolf optimization was implemented to extract the optimum number of disease blobs from mango leaf images. It was developed based on a hybrid metaheuristic model. To compress the actual image and to detect the infested spots, rescaling and contrast enhancement were performed before the feature extraction process. The leaf dataset used contains 450 numbers of leaves with four different kinds of classes in total. Among them, the diseased classes are gall midge, anthracnose, powdery mildew, and the last one is the healthy class. The extracted and selected features were fed as input to the ANN, and it classified effectively and achieved accuracy of 89.41%. This is better compared to other popular CNN approaches such as VGC, AlexNet, and ResNet-50. It requires a complete monitoring system (for all diseases), fine tuning parameters such as a number of layers and hidden nodes, activation function, and feature selection (the selection of optimal features from an image) [13].

Four different CNN models were implemented to detect plant leaf diseases, including Inceptionv3, InceptionResnetv2, MobileNetv2, and EfficientNetb0. In these, the standard convolution of the CNN was replaced with depth-wise separable convolution. The used dataset contained 54,305 number of leaf images with 38 categories of 14 different plant species which were taken from a plant image database. The same dataset with three different image formats, colored images, segmented images, and grayscale images, were trained and tested with the implemented CNN models. The parameters used to train the models were epochs, batch size, and dropout. The colored image format showed the results with high accuracy and less loss. This implies the better predictive ability of the implemented models in comparison with other deep learning models. Moreover, the MobileNetv2 model took less time to train the images with the optimized parameter. It can comfortably run on mobile phones [14].

The severity stages of citrus greening diseases were detected using different CNN models such as AlexNet, DenseNet169, Inception\_v3, ResNet34, SqueezeNet 1\_1, and VGG13. Among them, the Inception\_v3 model achieved the highest accuracy of 74.38% in 60 epochs. Augmentation was also used to increase the model's learning and performance. The process was performed using deep convolution Generative Adversarial Networks (GANs). The Inception\_v3 model achieved 20% higher accuracy with augmented data. The used data was collected from various non-profit organization websites such as Plant Village

and crowd AI; nonetheless, when deploying in this model real time, the result may have been influenced based on several attributes such as geographic area, sensor differences, cultivar, illumination differences, etc [15].

Three kinds of grape leaf spots are black measles, black rot, and leaf blight caused by several species of fungi identified by implementing the fine-grained GAN model. In order to detect and segment the leaf spot area from grape leaf images, Region-Based Convolutional Neural Networks (R-CNNs) were consolidated with fine-grained GAN. Then, the segregated sub parts were given as input to train the different classification models. ResNet-50 achieved the highest accuracy of 96.27% when predicting spots in grape leaves [16].

Grape leaf diseases are classified into four different classes, including three diseased classes and one healthy class. Initially, five state-of-the-art model evaluations were performed. The results imply that some counterpart structures performed well separately in terms of greater accuracy or reduced complexity. However, it took a long time to train them and occupied a large number of parameters. Among them, ShuffleNetV1 and ShuffleNetV2 results are balanced in both aspects of accuracy and complexity. Therefore, the lightweight CNN model was adopted, and then SEblocks (Squeeze and Excitation), were placed at the end of the three shuffle blocks. ShuffleNetV1 with a channel mechanism provided the enhanced visualization of important features, which reduced the computational cost and obtained the highest accuracy of 99.14% in a real-time grape disease classification task. The integration of suggested activity with an IoT platform would automate the process [17].

An improved DCNN was developed based on googlenet and cifar10 architectures to identify eight types of maize leaf diseases, including southern leaf blight, northern leaf blight, curvularia leaf spot, gray leaf spot, rust, dwarf mosaic, round spot, and brown spot. The Stochastic Gradient Descent (SGD) approach was used to perform parameter adjustment in the improved DCNN. To obtain substantial data, the 500maize dataset was collected from different sources such as Plant Village and other Google websites, which were augmented as 3060 images [18].

Maize leaves infected by armyworms were identified using remote sensing technologies and a CNN. The Shi-Tomasi algorithm was applied during pre-processing with the aim of detecting infected areas, including window panes and small holes, in a short time [19].

Four different types of white bean, red bean and pinto bean cultivars from three various species' leaf images were categorized using a discriminative CNN [20]. Both the backside and foreside of the foliage leaves were captured under controlled conditions used for training [21]. Two different loss functions, large margin cosine loss and additive angular margin loss, were employed together and separately in the CNN [22]. From the results, it was shown that the combined loss function produced better mean classification accuracy, and the result also implies that backside leaf images identify bean species and cultivars better than foreside images [23].

Emre ozbilge et al. proposed CNNs with data augmentation techniques to be employed during network training to improve the performance of the proposed network [24].

Piyush Juyale et al. showed that by using the R-CNN mask, they could improve and accelerate the pathogen detection process in plant leaves. Diseases could be identified earlier. The mean average precision value was 82% [25].

Prajwala TM et al. adopted a slight variation of the CNN model called LeNet to detect and identify diseases in tomato leaves. It achieved an average accuracy of 94–95%, indicating the feasibility of the neural network approach even under unfavorable conditions [26].

A comprehensive review of the state of the art of deep learning in biomedical applications by RyadZemouri et al. suggests that convolutional neural networks are one of the deep neural architectures that have replaced traditional machine learning methods. An analysis of the literature shows that CNNs have shown excellent abilities to transfer knowledge between different classification tasks performed by key deep neural architecture and weight transfer [27].

Geert Lidjens et al. reviewed key deep learning concepts related to medical image analysis and summarized over 300 contributions to the field which examined the application of deep learning to image classification, object detection, segmentation, registration, and other tasks. Finally, it was concluded that CNNs work perfectly compared with other classifications [28].

A large number of deep learning techniques implemented by many researchers are reviewed and discussed here. However, a limited number of research studies have focused on tomato crop disease detection, and there is still a need to improve the existing model. So, we proposed a CNN model with two convolution layers, two max pooling layers, a hidden layer, and a flattening layer for disease detection in tomato plants by analyzing images of leaves. Farmers no longer need to find crop specialists to solve the problems of diagnosing various crop diseases. Our model will help detect plant diseases early on, which will improve the quality and quantity of food crops and make them more profitable.

The rest of the paper is structured as follows: A discussion of the dataset and methodology is presented in Section 3. The experimental results and discussion are presented in Section 4, followed by the conclusion.

### 3. Methodology

#### 3.1. Dataset

The Plant Village dataset is a public dataset downloaded from Kaggle which contains 14 different types of crops. In this work, we chose the tomato leaf. There are 10 different classes in this tomato crop dataset. Nine of them are diseased classes, and one is a healthy class. This is shown in Figure 1. Disease class images with labels are shown in Figure 1. The number of images in each class is not in the same range. Some classes have images in the range of hundreds, and in some other classes, images are in the range of thousands. This kind of imbalanced dataset may affect the performance of a model. Therefore, for each class, 300 images were taken in this work. The data split ratio was fixed at 70:30 for training and testing, respectively, because empirical studies show that using 20–30% for testing and 80–20% for training enables a model to obtain the best result. The total number of images used for training and testing was 3000.

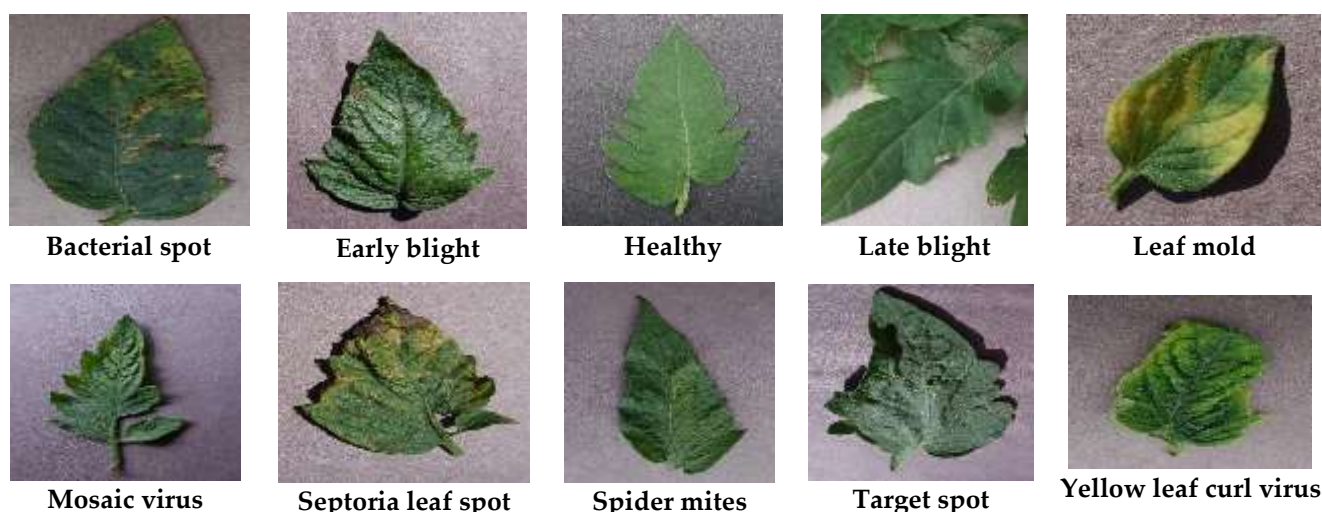


Figure 1. Different class sample images of tomato leaf dataset.

#### 3.2. Convolution Neural Network

The enhanced CNN model used in this work contained an input layer, two convolution layers, two max pooling layers, a hidden layer, and a flattening layer followed by an output layer, as shown in Figure 2.

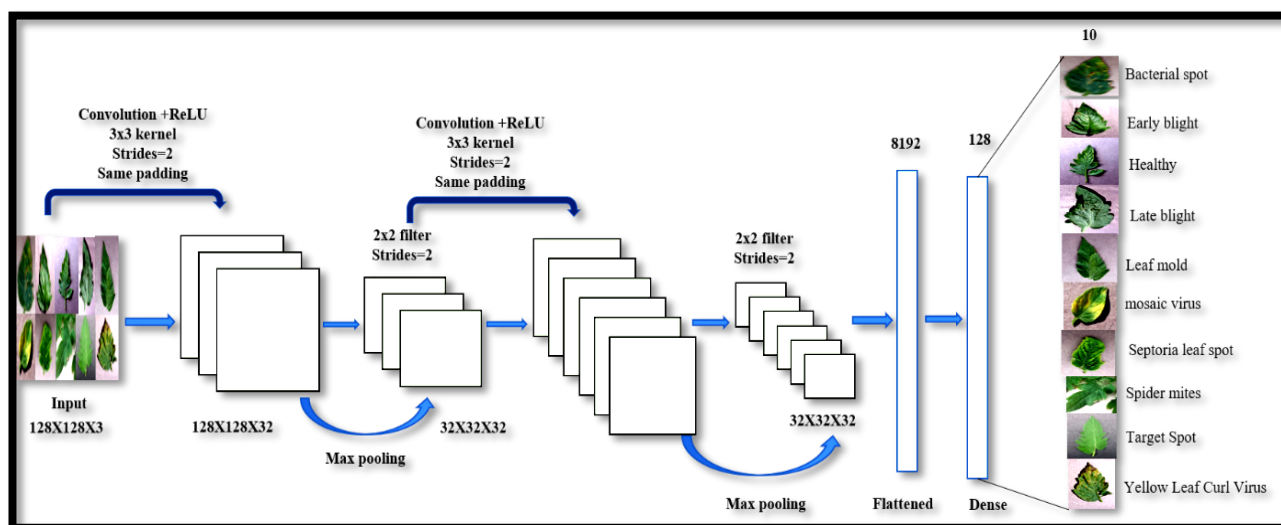


Figure 2. The Architecture of Convolutional Neural Network.

In the input layer, different types of tomato leaves were input. A padding process was applied to the image dataset to extract image information without losses. It enlarged the image by an extra pixel around the input image. Two different types of padding are commonly used: zero padding and neighboring pixel values. Here, zero padding was applied around the tomato leaf images. This helped us to obtain the same input size on the feature map. Thus, the features of entire input images could be processed without important feature loss. Then, convolution filters, also known as kernels, were applied to the padded input image to perform the convolution process. The size of the convolution filter was  $3 \times 3$  with strides of 1, and 32 different convolution filters were applied in both convolution layers.

Then, the convolution output was fed to the activation function. An activation function was used to introduce non-linearity into the network because it can recognize very complex patterns. In a CNN, it transforms the feature map values. There are many types of activation functions. The activation function used in the two-layer curve process wastrel (Rectified Linear Unit). A maximum pooling filter was applied over the feature maps. The architecture of Convolutional Neural Networks is shown in Table 1. The number of features extracted in the first convolution layer and the second convolution layer was 896 and 9248, respectively. The extracted tomato leaf image features were flattened in the flattening layer. Next, they were applied to the dense layer, also known as the fully connected layer.

Table 1. Structure of Convolutional Neural Network.

Layer(Type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 128, 128, 32)	0
max_pooling2d_2 (MaxPooling 2D)	(None, 64, 64, 32)	0
conv2d_3 (Conv2D)	(None, 64, 64, 32)	9248
max_pooling2d_3 (MaxPooling 2D)	(None, 32, 32, 32)	0
flatten_1 (Flatten)	(None, 8192)	0
dense_2 (Dense) 1048704	(None, 128)	0
dense_3 (Dense)	(None, 10)	(None, 10)
Total params: 1,060,138		
Trainable params: 1,060,138		
Non-trainable params: 0		

The number of neurons used in the fully connected layer was 128, and it extracted 1,048,704 features, while the output layer extracted 1290 features. The total number of extracted parameters was 1,060,138. All extracted parameters were trainable parameters, and there were zero non-trainable parameters. The SoftMax maximum activation function was used between the dense and output layers to classify 10 different tomato leaves.

## 4. Result and Discussion

### 4.1. Performance Analysis

Initially, the model was trained with different scaled input image sizes. The actual input data size was  $256 \times 256$ . Finally, the actual size of the input image was converted to  $128 \times 128$ , because the CNN model performance increased continuously with that input size. With other input quantities, the model was implemented without weight optimization during back propagation.

The number of image ratios in the tomato crop Plant Village dataset was different. All of the images were taken for the purpose of training and validation. The performance of the model was worse due to the imbalance of the datasets in each class. To make a balanced dataset, the total number of photographs shot in each class was 300. The data split ratio was 70:30, used for training and testing.

As shown in Table 2, the CNN model was implemented with different epochs of 100, 200, and 300. The test accuracy and loss of this model in 100 epochs were 0.944 and 1.0017, respectively. Then, in 200 epochs, the sample testing accuracy and loss were 0.6211 and 0.9758, respectively. When the model was trained with 300 epochs, it achieved test accuracy and loss of 0.9301 and 0.8478, respectively. The same result is shown graphically in Figure 3. From the abovementioned values, the model achieved high accuracy and low loss in 300 epochs during both training and validation.

**Table 2.** Training and validation accuracy and training and validation loss in CNN model.

Epoch	Training		Validation		Time/Step
	Loss	Accuracy	Loss	Accuracy	
100	0.9867	0.5671	1.0017	0.4944	19,289 ms
200	0.9612	0.6938	0.9758	0.6211	19,287 ms
300	0.9030	0.9857	0.9301	0.8478	28,419 ms

Then, the confusion matrix was generated to visualize the classification performance of the tomato leaf dataset using the CNN, as shown in Figure 3. The number of successfully classified images in 100 epochs during validation and training was 568 and 1636, respectively. The number of successfully classified images in 200 epochs during validation and training was 665 and 1855, respectively. During validation and training, there were 723 and 2083 correctly labeled images in 300 iterations, respectively.

Figure 4 is show that is a graphical representation of the performance of the CNN model.

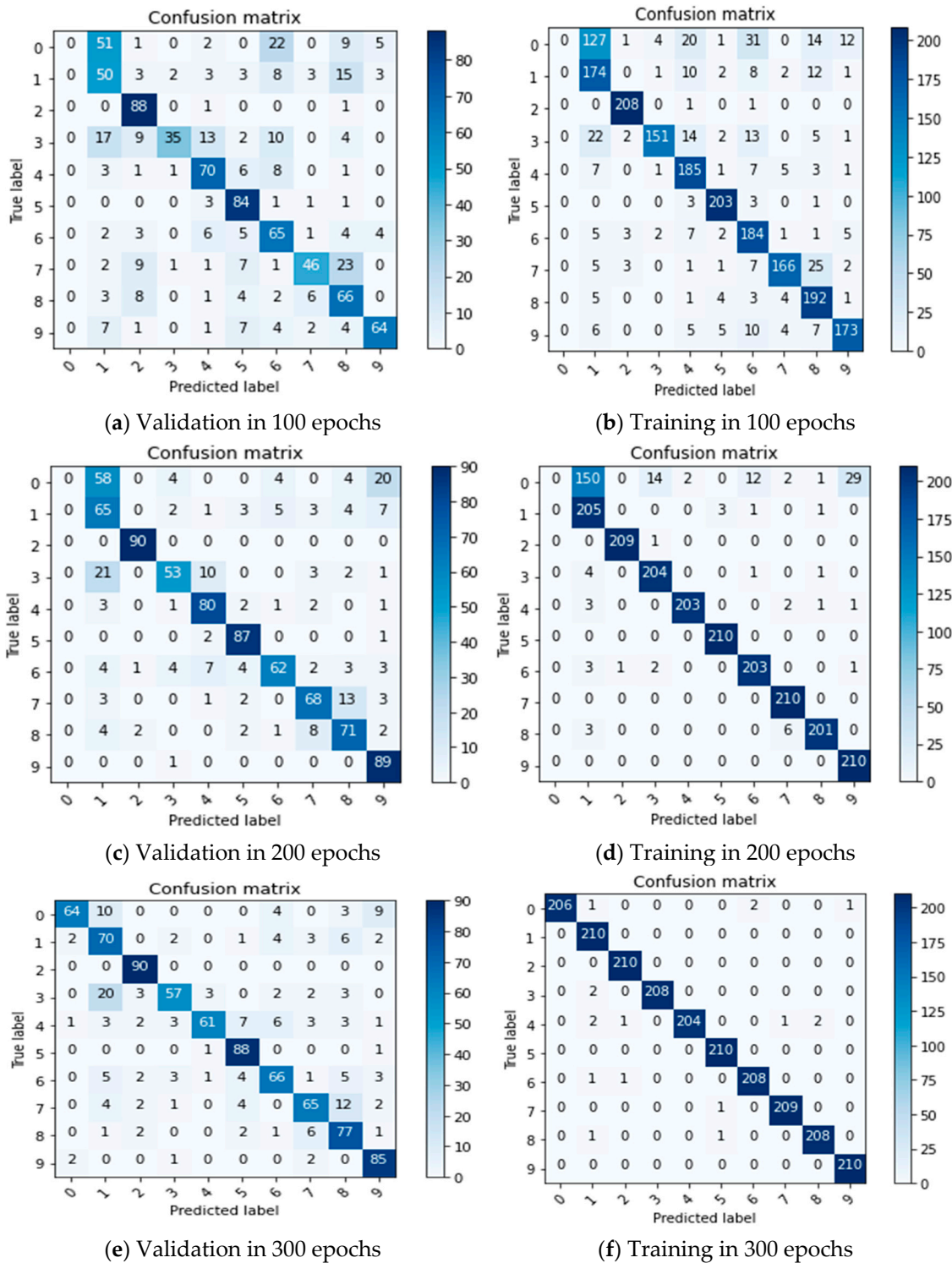
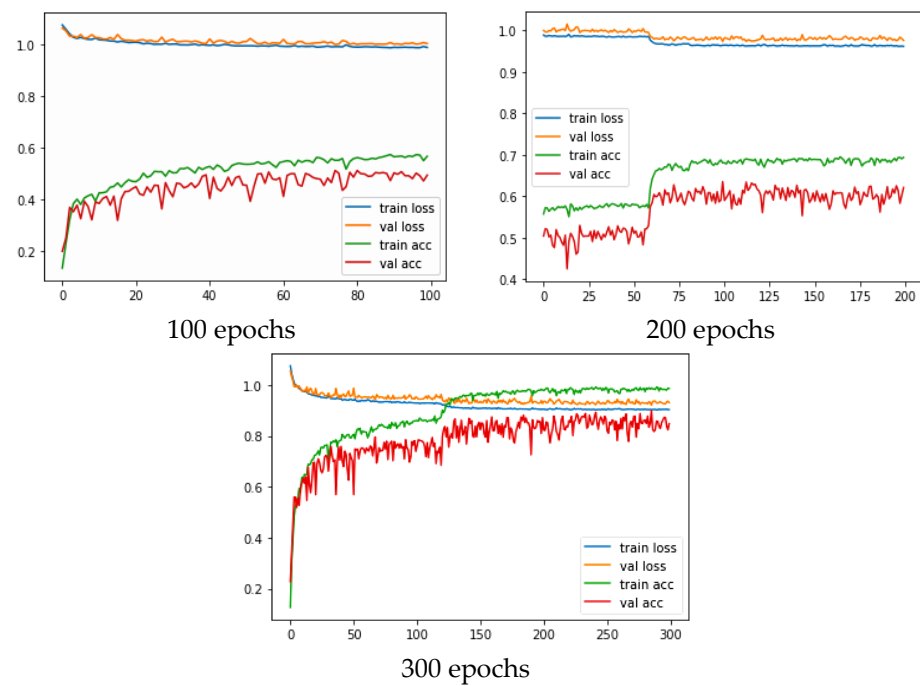


Figure 3. Confusion matrix of validation and training in different epochs.





**Figure 4.** Graphical representation of the performance of the CNN model.

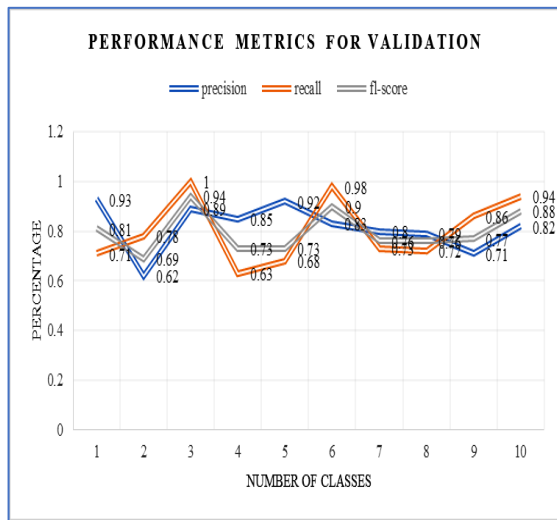
Apart from accuracy, some other matrices include recollection and f1 score macro averages and weighted averages. Accuracy alone is not enough to determine the performance of a CNN model. Therefore, these metrics were also evaluated, and the validation and training results for 300 epochs are shown in Table 3; Table 4. Precision, recall, and f1 score metrics were compared to find the best result. Figure 5a, b show a graphical representation of the comparison.

**Table 3.** Performance metrics in validation.

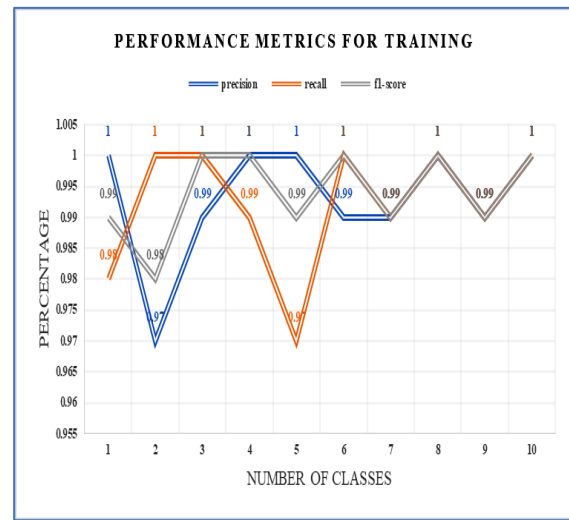
Classes	Precision	Recall	F1-Score	Support
0	0.93	0.71	0.81	90
1	0.62	0.78	0.69	90
2	0.89	1	0.94	90
3	0.85	0.63	0.73	90
4	0.92	0.68	0.73	90
5	0.83	0.98	0.9	90
6	0.8	0.73	0.76	90
7	0.79	0.72	0.76	90
8	0.71	0.86	0.77	90
9	0.82	0.94	0.88	90
accuracy			0.8	900
macro avg	0.82	0.8	0.8	900
weighted avg	0.82	0.8	0.8	900

**Table 4.** Performance metrics in training.

Classes	Precision	Recall	F1-Score	Support
0	1	0.98	0.99	210
1	0.97	1	0.98	210
2	0.99	1	1	210
3	1	0.99	1	210
4	1	0.97	0.99	210
5	0.99	1	1	210
6	0.99	0.99	0.99	210
7	1	1	1	210
8	0.99	0.99	0.99	210
9	1	1	1	210
accuracy			0.99	2100
macro avg	0.99	0.99	0.99	2100
weighted avg	0.99	0.99	0.99	2100



(a)



(b)

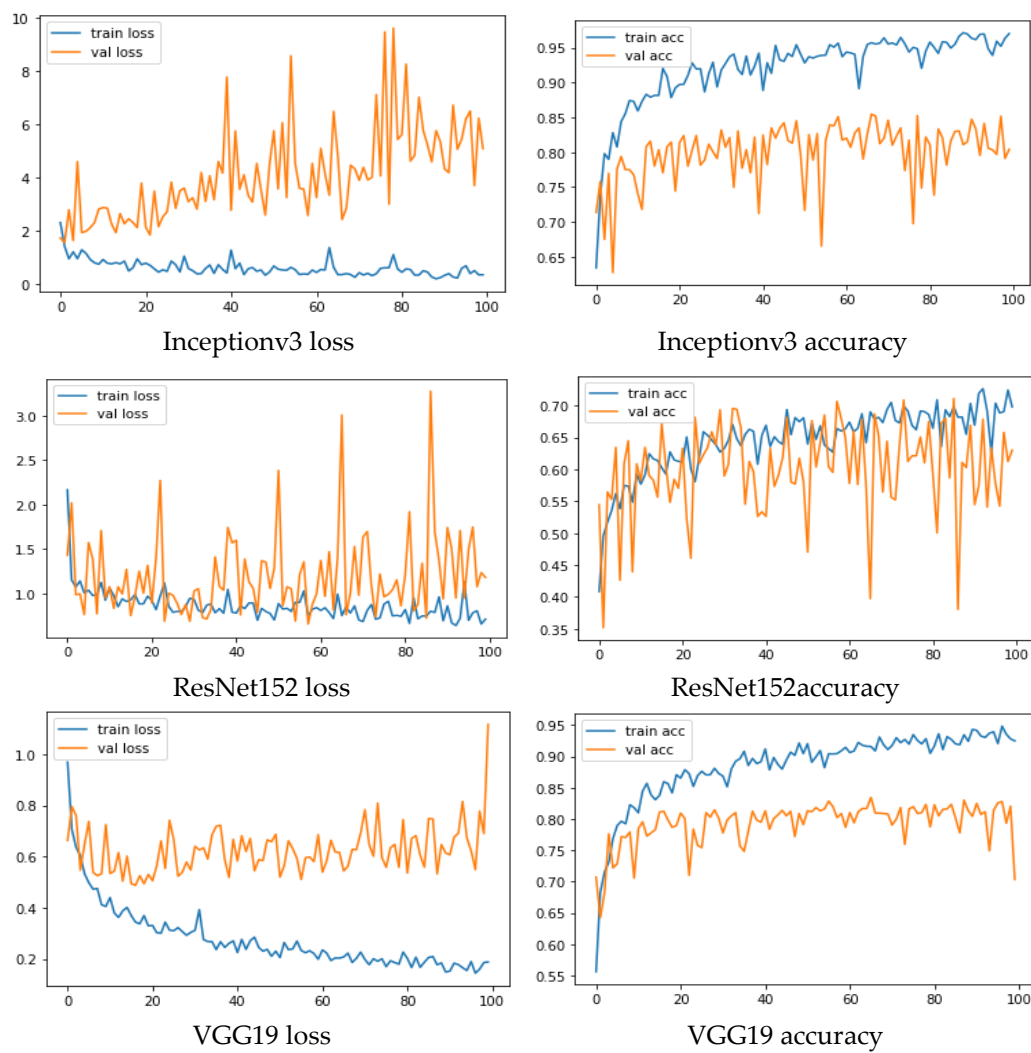
**Figure 5.** Graphical representation of validation and training accuracy.

4.2. Transfer Learning Techniques

Some transfer learning models (VGG19, ResNet152, and Inceptionv3 models) were trained and validated using the tomato leaf dataset. During the process, the models with retrained weights were used, and the final output layer count of the classes was changed from 100 to 10. The VGG19 model achieved a validation loss of 1.1144 and an accuracy of 0.7014. The ResNet152 model obtained a validation loss of 0.7968 and an accuracy of 0.6958. The inceptionV3 model obtained a loss of 5.1024 and an accuracy of 0.8024. Among them, the Inceptionv3 model achieved a high training accuracy of 0.9702 and a validation accuracy of 0.8024. The statistical values are shown in Table 5, and a graphical representation of the performance of each transfer learning technique is shown in Figure 6.

**Table 5.** Performance comparison of transfer learning techniques.

Transfer Learning	Training		Validation	
	Loss	Accuracy	Loss	Accuracy
VGG 19	0.1870	0.9256	1.1144	0.7014
ResNet 152	0.6556	0.7321	0.7968	0.6958
Inception V3	0.3632	0.9702	5.1024	0.8024

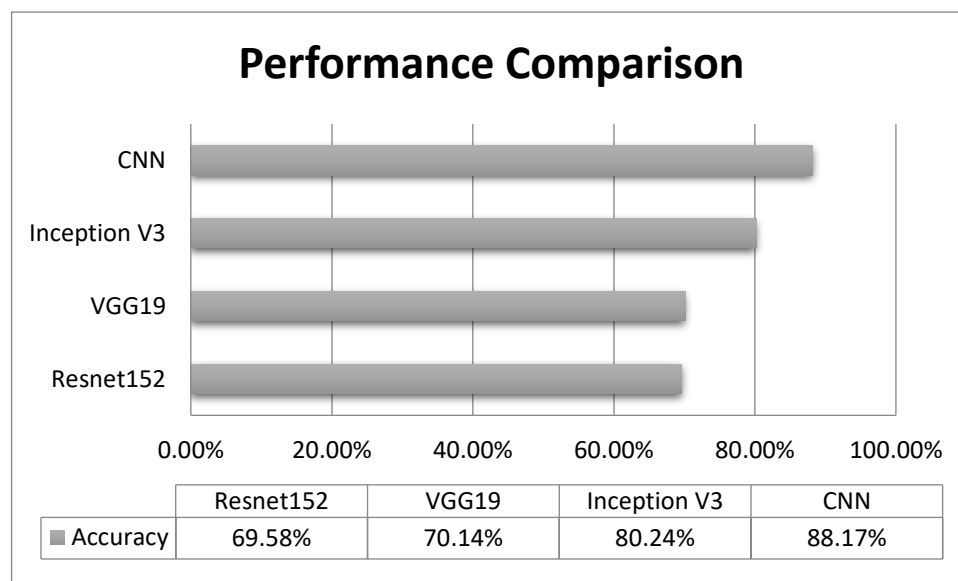


**Figure 6.** Graphical representation of transfer learning techniques.

#### 4.3. Comparison of Transfer Learning Techniques with CNN

In the Figure 7 shows the transfer learning techniques such as InceptionV3, ResNet 152, VGG 19 and their results were compared with our proposed CNN model. Among them, the ResNet152 model achieved the lowest validation accuracy of 69.58%.

The VGG19 model achieved a validation accuracy of 70.14%, and the InceptionV3 model achieved a validation accuracy of 80.24%. While the VGG19 model and the ResNet 152 model did better, with an accuracy of over 70%, the proposed CNN model did even better, with an accuracy of 88.17%.



**Figure 7.** Comparison of transfer learning techniques with CNN.

## 5. Conclusions

Tomato leaf disease detection and categorization employ a variety of deep learning approaches. Compared to other transfer learning techniques such as ResNet152, VGG19, and InceptionV3, the CNN model performed well in terms of disease detection in tomato crops. The proposed model has a training accuracy of 98% and a testing accuracy of 88.17%. Farmers can actually overcome their plant identification difficulties without the need to follow plant scientists. This will help them to cure tomato plant diseases in time, so they can improve the quality and quantity of their tomato crops, as well as profit. In the future, we hope to improve the model with a different crop. Additionally, we will try to optimize the same model on the same dataset to further improve the test accuracy.

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

**Funding:** This research received no external funding.

**Data Availability Statement:** The data used to support the findings of this study are available from the corresponding author upon request.

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

## References

- Hassan, S.M.; Maji, A.K.; Jasiński, M.; Leonowicz, Z.; Jasińska, E. Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach. *Electronics* **2021**, *10*, 1388. [\[CrossRef\]](#)
- Habiba, S.U.; Islam, M.K. Tomato Plant Diseases Classification Using Deep Learning Based Classifier From Leaves Images. In Proceedings of the 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), Dhaka, Bangladesh, 27–28 February 2021; pp. 82–86.
- Bedi, P.; Gole, P. Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. *Artif. Intell. Agric.* **2021**, *5*, 90–101. [\[CrossRef\]](#)
- Fazari, A.; Pellicer-Valero, O.J.; Gómez-Sanchís, J.; Bernardi, B.; Cubero, S.; Benalia, S.; Zimbalatti, G.; Blasco, J. Application of deep convolutional neural networks for the detection of anthracnose in olives using VIS/NIR hyperspectral images. *Comput. Electron. Agric.* **2021**, *187*, 106252. [\[CrossRef\]](#)
- Sambasivam, G.; Opiyo, G.D. A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks. *Egypt. Inform. J.* **2021**, *22*, 27–34. [\[CrossRef\]](#)

6. Afonso, M.; Blok, P.M.; Polder, G.; van der Wolf, J.M.; Kamp, J. Blackleg Detection in Potato Plants using Convolutional Neural Networks. This project was funded by: Branche Organisation Arable Farming (BO-Akkerbouw), Dutch Farmers Organisation (LTO-Nederland), Kverneland Mechatronics, Agrico, HZPC, NAK and the Dutch Topsector AgriFood "Op naar precisielandbouw 2.0" (AF-14275). *IFAC-PapersOnLine* **2019**, *52*, 6–11. [[CrossRef](#)]
7. Wang, F.; Wang, R.; Xie, C.; Zhang, J.; Li, R.; Liu, L. Convolutional neural network based automatic pest monitoring system using hand-held mobile image analysis towards non-site-specific wild environment. *Comput. Electron. Agric.* **2021**, *187*, 106268. [[CrossRef](#)]
8. Picon, A.; Seitz, M.; Alvarez-Gila, A.; Mohnke, P.; Ortiz-Barredo, A.; Echazarra, J. Crop conditional Convolutional Neural Networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions. *Comput. Electron. Agric.* **2019**, *167*, 105093. [[CrossRef](#)]
9. Zhang, S.; Zhang, S.; Zhang, C.; Wang, X.; Shi, Y. Cucumber leaf disease identification with global pooling dilated convolutional neural network. *Comput. Electron. Agric.* **2019**, *162*, 422–430. [[CrossRef](#)]
10. Ashraf, R.; Habib, M.A.; Akram, M.U.; Latif, M.A.; Malik, M.S.; Awais, M.; Dar, S.H.; Mahmood, T.; Yasir, M.; Abbas, Z. Deep Convolution Neural Network for Big Data Medical Image Classification. *IEEE Access* **2020**, *8*, 105659–105670. [[CrossRef](#)]
11. Reddy, T.V.; Rekha, K.S. Deep Leaf Disease Prediction Framework (DLDPF) with Transfer Learning for Automatic Leaf Disease Detection. In Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 8–10 April 2021; pp. 1408–1415. [[CrossRef](#)]
12. Mishra, S.; Sachan, R.; Rajpal, D. Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition. *Procedia Comput. Sci.* **2020**, *167*, 2003–2010. [[CrossRef](#)]
13. Pham, T.N.; Tran, L.V.; Dao, S.V.T. Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection. *IEEE Access* **2020**, *8*, 189960–189973. [[CrossRef](#)]
14. Singh, J.; Kaur, H. Plant disease detection based on region-based segmentation and KNN classifier. In Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018, Palladam, India, 16–17 May 2018; pp. 1667–1675.
15. Zeng, Q.; Ma, X.; Cheng, B.; Zhou, E.; Pang, W. GANs-Based Data Augmentation for Citrus Disease Severity Detection Using Deep Learning. *IEEE Access* **2020**, *8*, 172882–172891. [[CrossRef](#)]
16. Zhou, C.; Zhang, Z.; Zhou, S.; Xing, J.; Wu, Q.; Song, J. Grape Leaf Spot Identification Under Limited Samples by Fine Grained-GAN. *IEEE Access* **2021**, *9*, 100480–100489. [[CrossRef](#)]
17. Tang, Z.; Yang, J.; Li, Z.; Qi, F. Grape disease image classification based on lightweight convolution neural networks and channelwise attention. *Comput. Electron. Agric.* **2020**, *178*, 105735. [[CrossRef](#)]
18. Zhang, X.; Qiao, Y.; Meng, F.; Fan, C.; Zhang, M. Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks. *IEEE Access* **2018**, *6*, 30370–30377. [[CrossRef](#)]
19. Ishengoma, F.S.; Rai, I.A.; Said, R.N. Identification of maize leaves infected by fall armyworms using UAV-based imagery and convolutional neural networks. *Comput. Electron. Agric.* **2021**, *184*, 106124. [[CrossRef](#)]
20. Tavakoli, H.; Alirezazadeh, P.; Hedayatipour, A.; Nasib, A.H.B.; Landwehr, N. Leaf image-based classification of some common bean cultivars using discriminative convolutional neural networks. *Comput. Electron. Agric.* **2021**, *181*, 105935. [[CrossRef](#)]
21. Patel1, S.; Jaliya1, U.K.; Patel1, P. A Survey on Plant Leaf Disease Detection. *Int. J. Mod. Trends Sci. Technol.* **2020**, *6*, 129–134.
22. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Thirty-First AAAI Conference on Artificial Intelligence. Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv* **2016**, arXiv:1602.07261. [[CrossRef](#)]
23. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [[CrossRef](#)]
24. Özbilge, E.; Ulukök, M.K.; Toygar, Ö.; Ozbilge, E. Tomato Disease Recognition Using a Compact Convolutional Neural Network. *IEEE Access* **2022**, *10*, 77213–77224. [[CrossRef](#)]
25. Juyal, P.; Sharma, S. Detecting the Infectious area along with Disease using Deep Learning in tomato Plant Leaves. In Proceedings of the 3rd International Conference on Intelligent Sustainable Systems (ICISS), Thoothukudi, India, 3–5 December 2020. [[CrossRef](#)]
26. Tm, P.; Pranathi, A.; SaiAshritha, K.; Chittaragi, N.B.; Koolagudi, S.G. Tomato Leaf Disease Detection using Convolutional Neural Networks. In Proceedings of the Innovations in Intelligent Systems and Applications Conference (ASYU), Coimbatore, India, 10–12 June 2020; pp. 1–5. [[CrossRef](#)]
27. Zemouri, R.; Zerhouni, N.; Racoceanu, D. Deep Learning in the Biomedical Applications: Recent and Future Status. *Appl. Sci.* **2019**, *9*, 1526. [[CrossRef](#)]
28. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; van der Laak, J.A.W.M.; van Ginneken, B.; Sánchez, C.I. A survey on deep learning in medical image analysis. *Med. Image Anal.* **2017**, *42*, 60–88. [[CrossRef](#)] [[PubMed](#)]