J. Hortic. Sci.
Vol. 20(2), 2025

https://doi.org/10.24154/jhs.v20i2.3321

Original Research Paper

Performance evaluation of a custom convolutional neural network for diagnosing mite infestation and diseases in tomato (*Solanum lycopersicum* L.)

Gulati A[*], Singh A.P., Chug A. and Ahlawat K.
University School of Information, Communication & Technology
Guru Gobind Singh Indraprastha University, New Delhi - 110 078, India
[*]Corresponding author Email: amulyagulati@gmail.com

ABSTRACT

A customized convolutional neural network (CNN) model is developed and fine-tuned for identifying tomato leaf diseases. The model is optimized by adjusting hyperparameters, batch size, and CNN layers. It is tested against mite infestation and ten diseases, including one bacterial, two viral, and seven fungal diseases. Compared to other models, customized CNN based model performed well both in terms of accuracy and execution time. Performance was analysed using loss-accuracy graphs and a confusion matrix. Evaluation metrics for test images from the original dataset showed an average accuracy, precision, recall, and F1 score of 99.64 per cent, with datasets for bacterial spot, leaf curl virus, leaf mould, Cercospora leaf spot, mosaic virus, and verticillium wilt achieving 100 per cent in these metrics. Dataset for two-spotted mite infestations also showed 100 per cent accuracy in recognising the damage. The execution time of custom CNN model on the tomato leaf disease dataset averaged 1339.09 seconds after 25 epochs, 1356.91 seconds after 50 epochs, and 2696 seconds in total for training across mite infestation and ten disease classes. The model accurately identified all 595, 896, 266, 56, 105, 575 diseased images of bacterial spot, leaf curl, leaf mould, Cercospora leaf spot, mosaic virus, verticillium wilt and 1981 images of two-spotted mite infested leaves, as well as 448 healthy images. These results demonstrate that the customized CNN model is highly effective and efficient in diagnosing mite infestation and a variety of tomato leaf diseases, offering a reliable tool for disease management.

Keywords: Accuracy, convolutional neural network model, execution time, tomato leaf diseases

INTRODUCTION

Tomato plants (*Solanum lycopersicum*) are susceptible to a variety of diseases caused by fungi, bacteria, viruses, and other pathogens, which can significantly diminish both crop yield and quality (Sonika et al., 2017). These diseases present substantial challenges for farmers and pose a threat to global food security. As leaves are key indicators of plant health, developing a method to detect leaf diseases by analysing the characteristics of spots on tomato leaves such as shape, colour, and orientation is crucial for enhancing agricultural nutrition programs (Shruthi et al., 2022).

Historically, image processing paired with data mining techniques such as K-nearest neighbour (KNN), backpropagation neural networks, support vector machines (SVM), and spatial gray-level dependency matrices have been employed to identify plant diseases (Afifi et al., 2020; Choudhary & Hiranwal, 2021). However, recent advances in artificial intelligence, especially in deep learning, have ushered in new capabilities for automating the detection and diagnosis of plant diseases. Among these techniques, Convolutional Neural Networks (CNNs) have become a prominent method for classification and feature extraction across various fields including natural language processing, speech recognition, and computer vision (Shoaib et al., 2023; Gulati et al., 2024). This study focuses on the development and optimization of a customized CNN model tailored for diagnosing diseases in tomato leaves.

MATERIALS AND METHODS

Data acquisition

To ensure a robust and diverse dataset, we sourced data from publicly accessible datasets such as Plant Disease Expert and Plant Village, available on https://www.kaggle.com/PlantVillage, alongside datasets from agricultural universities' websites and research publications. This comprehensive dataset included high-resolution images of both healthy and diseased tomato plants, with an example shown in Fig. 1.

Tomato Healthy leaf

Tomato Bacterial spot

Tomato Early blight

Tomato late blight

Tomato leaf curl virus

Tomato Leaf mould

Tomato Cercospora leaf spot

Tomato Mosaic virus

Tomato target spot

Tomato Septoria leaf spot

Tomato Two spotted mite

Tomato Verticilium wilt

Fig. 1 : Diseased and healthy representative leaf images of tomato crop

Data pre-processing

All images were resized to a uniform dimension of 112 x 112 pixels to standardize the input for the deep learning model. To facilitate model convergence and avoid numerical instability, pixel values were normalized to a scale ranging from 0 to 1. This normalization ensures that all features contribute equally during the learning process.

Table 1 : Data augmentation techniques

Technique	Value	Purpose
Horizontal Flip		It simulates the scenario of viewing the crop from different side
Vertical Flip		It simulates the scenario of viewing the crop from different side
Rotate 45°	45°	It helps the model to recognize diseases in images taken from different orientation
Rotate 60°	60°	It helps the model to recognize diseases in images taken from different orientation
Scaling	1.2	It refers to zooming in or out of the image, which can be a useful augmentation for improving the robustness of model
Shearing	0.2	It helps the model learn to recognize objects in images that are distorted or regardless of their size.

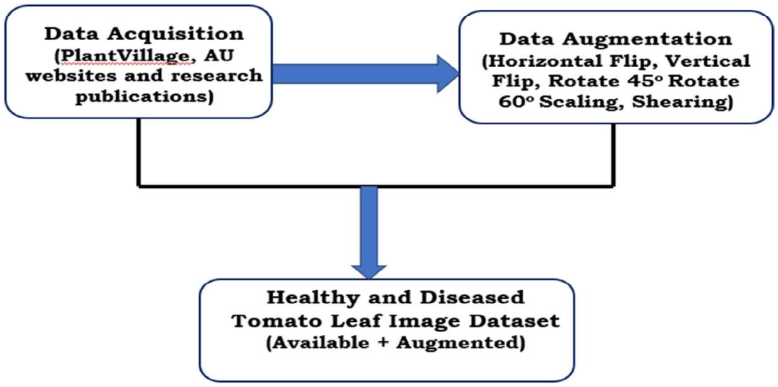


Fig. 2 : Flow chart for disease data set generation

Data augmentation

To ensure model generalization, six sophisticated data augmentation techniques were applied to the training dataset, which included both diseased and healthy tomato leaf images (Table 1). These techniques significantly expanded the dataset to many times its original size, helping to prevent overfitting. This expansion resulted in a more robust and diverse dataset, ultimately enhancing the performance of the machine learning model. The disease data set generation and working process of tomato disease recognition is explained in Fig. 2.

Optimization technique

The adaptive moment estimation (Adam) optimizer was utilized, which calculates adaptive learning rates for each parameter. Adam enhances optimization by using exponentially moving averages of the gradient from the current minibatch to estimate the moments. This approach has shown significant improvements over traditional methods such as regular and stochastic gradient descent.

Data annotation

Data annotation involves labelling the individual elements within the training data to clarify their content and significance. This process included the addition of contextual information that machine learning models utilize for learning. The annotated data was subsequently used for training the model.

Data partition

All images were categorized by class numbers and labels, distinguishing between healthy and diseased states. These categories were then split into training

Table 2 : Class labels and statistics of tomato leaf dataset

Class number	Class	No. of original images	No. of augmented images	Training images	Testing images	Total images
1	Healthy	318	1908	1778	448	2226
2	Bacterial spot	425	2550	2380	595	2975
3	Early blight	200	1200	1120	280	1400
4	Late blight	382	2100	1950	532	2482
5	Leaf curl virus	642	3852	3598	896	4494
6	Leaf mould	190	1140	1064	266	1330
7	Cercospora leaf spot	38	228	210	56	266
8	Mosaic virus	75	450	420	105	525
9	Target spot	281	1686	1575	392	1967
10	Septoria leaf spot	354	2124	1981	497	2478
11	Two spotted mite	1417	8502	7938	1981	9919
12	Verticilium wilt	773	2094	2294	575	2867
13	Total images	5095	27834	26306	6623	32929

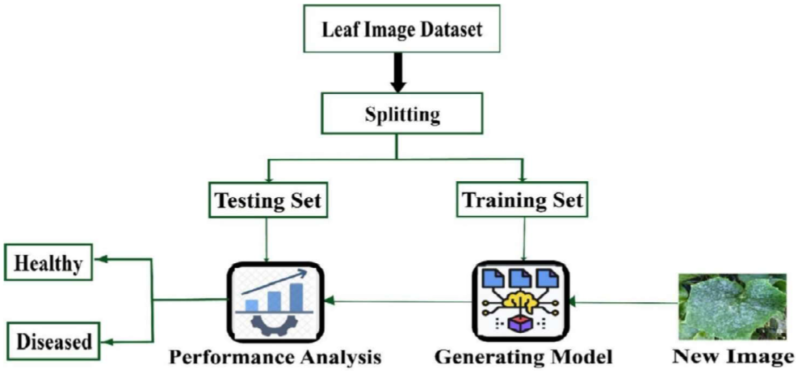


Fig. 3: Flow chart for working process of tomato crop disease recognition

and testing datasets (Table 2). All the images were classified into class numbers and class labels (healthy and disease) which were further divided into training and test datasets (Table 2).

Model development

A custom convolutional neural network (CNN) was developed by adjusting batch sizes, pre-processing techniques, and the number of convolutional layers to enhance its efficacy on both healthy and diseased tomato leaf image datasets. Model 1 was developed initially by keeping batch size value as 16, number of epochs as 100 and number of convolutional layers as 3. In the second model (model 2), the batch size was doubled (32) and epochs were reduced to half (50) with no change in convolutional layers. To further increase the efficiency, model 3 was tested with same batch size and epochs as in model 2 but number of convolutional layers was increased to 4 with minor change in rotation, zoom, height and width. In model 5, one more convolutional layer was added. The workflow for the tomato crop disease recognition process is illustrated in Fig. 3.

Implementation

The proposed method was implemented using Python. The adjustment of the learning rate for each epoch was governed by the following equation:

$$\propto_n = \propto_0 * 0.4 \, \frac{1 + epoch(n)}{4}$$

Where α_n, α_0 and epoch(n) represent the learning rate at n^{th} epoch, initial learning rate, and the current epoch number respectively.

Loss accuracy graphs

The loss graph in machine learning tracks the model's error; the discrepancy between the predicted and actual values throughout the training process. This graph helps to visualize the model's performance, with the objective of minimizing loss during training. Conversely, the accuracy graph charts how the model's accuracy evolves over training sessions. Here, the x-axis indicates the training epochs, while the y-axis shows the accuracy levels achieved by the model.

Performance evaluation of classification algorithm

The performance of the classification algorithm was evaluated using a Confusion Matrix, which provides a detailed account of the model's correct and incorrect classifications across a specific dataset. To thoroughly assess model performance, metrics such as accuracy, precision, recall, and F1-score were computed.

$$P_a = \frac{TP_a}{TP_a + FP_a}, \qquad R_a = \frac{TP_a}{TP_a + FN_a},$$

$$F1_a = 2 \times \frac{P_a \times R_a}{P_a + R_a}, \quad ACC = \frac{TP_a + TN_a}{TP_a + TN_a + FP_a + FN_a}$$

Where TP_a, TN_a, FP_a and FN_a represent true positive, true negative, false positive and false negative for class 'a'. Similarly, P_a, R_a $F1_a$ and ACC represent precision, recall, F1-score and accuracy for class 'a'. Class 'a' represents dataset of particular disease or mite infested images.

Correct identification was calculated from confusion matrix of mite infested dataset and diseased dataset of ten diseases by applying formula:

$$Correct\ Identification\ (\%) = \frac{Correctly\ Identified\ Images}{Total\ Images} \times 100$$

RESULTS AND DISCUSSION

Convolutional neural networks (CNNs) are one of the most important algorithms for image classification which does not need any manual feature extraction to work and makes it robust against new datasets (Lecun et al., 1998). To correctly train the CNN to be able to classify images, many hyperparameters need to be adjusted; these hyperparameters affect the performance of the network along its time to convergence. The techniques and hyperparameters selected in developing CNN model are discussed.

Dataset statistics

The compilation of datasets for tomato crop diseases revealed the original collection of 318 healthy leaf images and 4,777 diseased leaf images spanning several conditions including bacterial spot, early blight, late blight, leaf curl virus, and others (Table 2). From publicly available sources, these datasets were expanded through data augmentation to include 5,095 originals and 27,834 augmented images. Image augmentation is usually used to increase the image dataset and also to make the network more robust against translation invariance. During present study, six advanced data augmentation techniques; horizontal flip, vertical flip, rotate 45° rotate 60° scaling and shearing were applied to mite infested and disease images of tomato leaves. CNN model generates better results when they are trained using a larger dataset (Saini & Seba, 2019), whereas, if the model is trained on a smaller dataset with few training samples, it may perform badly during validation and testing as it memorizes irrelevant noise instead of detecting significant discriminative features (Cagli et al., 2017). Augmented data presents a more comprehensive set of possible points which minimized the distance between training, validation and test samples (Shorten & Khoshgoftaar, 2019). Experimental results of Mikołajczyk and Grochowski (2018), and Poojary et al. (2021) also showed that their models achieved better test accuracy when data augmentation was employed.

Model development for tomato diseases

The initial custom CNN model utilized a batch size of 16, yielding an accuracy of 0.58, precision of 0.65, recall of 0.58, and F1 score of 0.57 (Table 3). Adjusting the batch size to 32 improved these metrics substantially, increasing all scores to 0.81 after reducing the training epochs by half. Batch size and number of epochs are another important hyperparameters in the development of model. It was interesting to note that in the initial model when the batch size was 16, the accuracy of the model was 0.58 even after 100 epochs. When the batch size was doubled, it significantly improves the accuracy to 0.81 after half the epochs (50). Batch size is the number of images used to train a single forward and backward pass. On one hand, a small batch size can converge faster than a large batch, but a large batch can reach optimum minima that a small batch size cannot reach. Radiuk (2017) investigated the effect of batch size on CNN performance for image classification and concluded that the higher the batch size the higher the network accuracy, meaning that the batch size has a huge impact on the CNN performance. However, many authors stated that a

Table 3 : Development and performance of custom CNN model for tomato diseases

Parameter	Model 1	Model 2	Model 3	Model 4
Batch size	16	32	32	32
Epochs	100	50	50	50
Random rotation	(0.2)	(0.2)	(0.3)	(0.3)
Random zoom	(0.2)	(0.2)	(0.3)	(0.3)
Random height	(0.2)	(0.2)	(0.3)	(0.3)
Random width	(0.2)	(0.2)	(0.3)	(0.3)
Number of convolutional layers	3	3	4	5
Performance of CNN model				
Accuracy	0.58	0.81	0.85	0.87
Precision	0.65	0.81	0.85	0.86
Recall	0.58	0.81	0.85	0.87
F1 Score	0.57	0.81	0.85	0.86

batch size of 32 is a good default value, which is more robust and optimal to quicken the computation of the network and help the network achieve the highest accuracy in the shortest time (Bengio, 2012; Masters & Luschi, 2018; Kandel & Castelli, 2020). These results were in conformity with the present work.

A single pass over the complete training dataset constitutes an epoch in deep learning. The model uses backpropagation to update its parameters throughout each epoch in order to lower the loss function. Accuracy tends to increase with the number of epochs, as the model continues to refine its understanding of the training data. However, after a certain point, increasing the number of epochs can lead to overfitting, where the model becomes too focused on the training data and performs poorly on new, unseen data. Finding the ideal number of epochs that strikes a balance between underfitting and overfitting can be achieved by employing strategies like early stopping (Vinayedula, 2023). In present case, 50 epochs gave better accuracy.

Further, enhancements were achieved by adding an additional convolutional layer, leading to an accuracy, precision, recall, and F1 score of 0.85 with 4 convolutional layers. This performance slightly increased to an accuracy and F1 score of 0.87 with five convolutional layers, alongside precision and recall rates of 0.86 (Table 3). A convolutional layer is the main building block of the CNN model, which contains a set of kernels learned throughout the training process. Comparative analysis shows that this model outperformed similar CNNs reported in previous studies by Foysal et al. (2020) and Sardogan et al. (2018), which gave 76 and 86 percent accuracy. This study approached the high benchmarks set by Brahimi et al. (2017), which reported 94.53 percent accuracy using CNN compared with shallow models and handcrafted features and Abbas et al. (2021), who achieved higher accuracies (97.11%) than the present study model by utilizing CGAN algorithm with pre trained DensNet121 model.

Performance evaluation

The custom CNN model's evaluation on the tomato disease dataset, which includes 11 distinct class labels, was comprehensively assessed using various metrics (Table 4), execution times (Table 5), loss-accuracy graph (Fig. 5), and a confusion matrix (Fig. 5). True and predicted results of diseased and healthy leaf images are presented in Fig. 5.

Test images from the original dataset demonstrated exemplary performance, with an average score of 99.64 per cent across accuracy, precision, recall, and F1 score metrics, with several conditions achieving perfect scores (Table 4).

The model's performance of our model across mite infestation and 10 disease classes was better (99.51%) than the performance of the 2 (99.01%, Emebo et al., 2019), 5 (96.6%, Widiyanto et al., 2019), 7 (97.49%, Rangarajan et al., 2018), 9 (99.18%, Brahimi et al., 2017; 97.28%, Zhang et al., 2018) and 10 (91.2%, Agarwal et al., 2020; 99.12%, Afify et al., 2022) class classification task.

Execution time

The training of the custom CNN model on the tomato leaf disease dataset required 1,339.09 seconds for 25 epochs, 1,356.91 seconds for 50 epochs, and a total of 2,696 seconds across all training phases (Table 5). This efficiency compares favourably with more time-intensive models like AlexNet (3360 s), VGG16 (3390 s) after 30 epochs, DenseNet (4280 s), and ResNet (2420 s) after 20 epochs on the identification of tomato leaf diseases (Bouni et al., 2023).

Loss and accuracy trends

Initially, the model exhibited its highest loss at epoch zero, which progressively decreased with additional epochs, indicating fewer prediction errors. The accuracy graphs (Fig. 3) reflected a steady improvement in the model's ability to correctly identify diseases as training progressed.

Confusion matrix analysis

The confusion matrix for the custom CNN model demonstrated its capability to accurately identify all instances of bacterial spot, leaf curl, leaf mould, Cercospora leaf spot, mosaic virus, two-spotted mite, verticillium wilt disease, and healthy tomato leaves across 595, 896, 266, 56, 105, 1981, 575 diseased images, and 448 healthy images respectively (Fig. 5 a, d, e, f, g, j, k). The true and predicted results are clearly presented in Fig. 5. For early blight, late blight, target spot, and Septoria leaf spot-infested leaves totalling 285, 532, 396, and 497 images; the model achieved high accuracy rates, correctly identifying 280 (98.24%), 526 (98.87%),

Table 4 : Evaluation metrics for test images from the original dataset

Class	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Correct identification (%)
Bacterial spot	100	100	100	100	100
Early blight	99	99	99	99	98.24
Late blight	99	99	99	99	98.87
Leaf curl virus	100	100	100	100	100
Leaf mold	100	100	100	100	100
Cercospora leaf spot	100	100	100	100	100
Mosaic virus	100	100	100	100	100
Target spot	99	99	99	99	98.73
Septoria leaf spot	99	99	99	99	98.79
Two spotted mite	100	100	100	100	100
Verticilium wilt	100	100	100	100	100
Average score	99.64	99.64	99.64	99.64	99.51

Table 5 : Execution time of custom CNN model on tomato disease leaf dataset

Class	Total parameters	Size (MB)	Execution time(s)		
			After 25 epochs	After 50 epochs	Total
Bacterial spot	651842	2.49	1307	1355	2662
Early blight	651842	2.49	823	838	1661
Late blight	651842	2.49	1242	1303	2545
Leaf curl virus	651842	2.49	1884	2401	4285
Leaf mold	651842	2.49	1031	1002	2033
Cercospora leaf spot	651842	2.49	671	599	1270
Mosaic virus	651842	2.49	741	618	1359
Target spot	651842	2.49	1043	887	1930
Septoria leaf spot	651842	2.49	1306	1245	2551
Two spotted mite	651842	2.49	3267	3383	6650
Verticilium wilt	651842	2.49	1415	1295	2710
Average score			1339.09	1356.91	2696

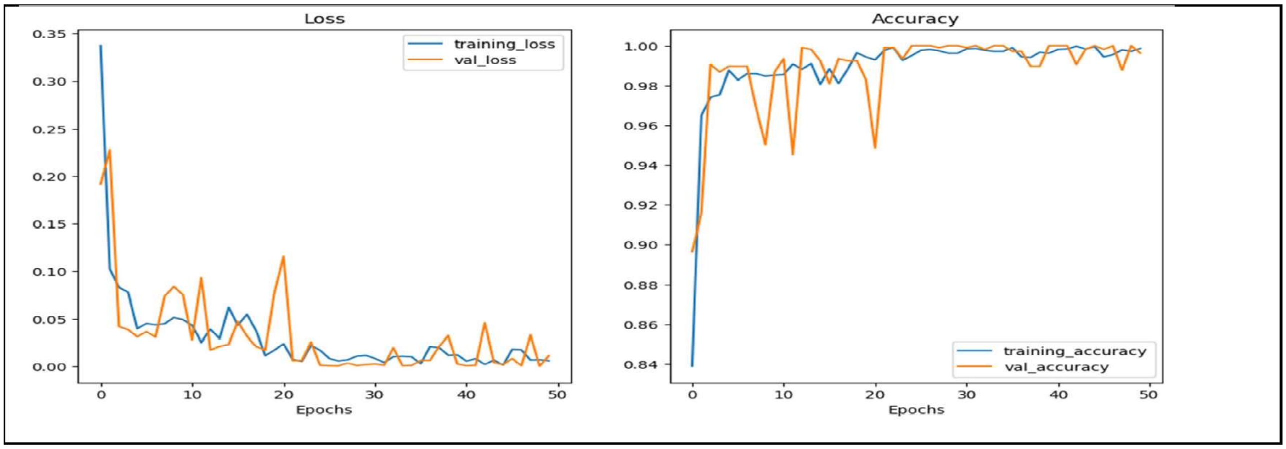


Fig. 4 : Loss-accuracy graph for Custom CNN model on disease dataset

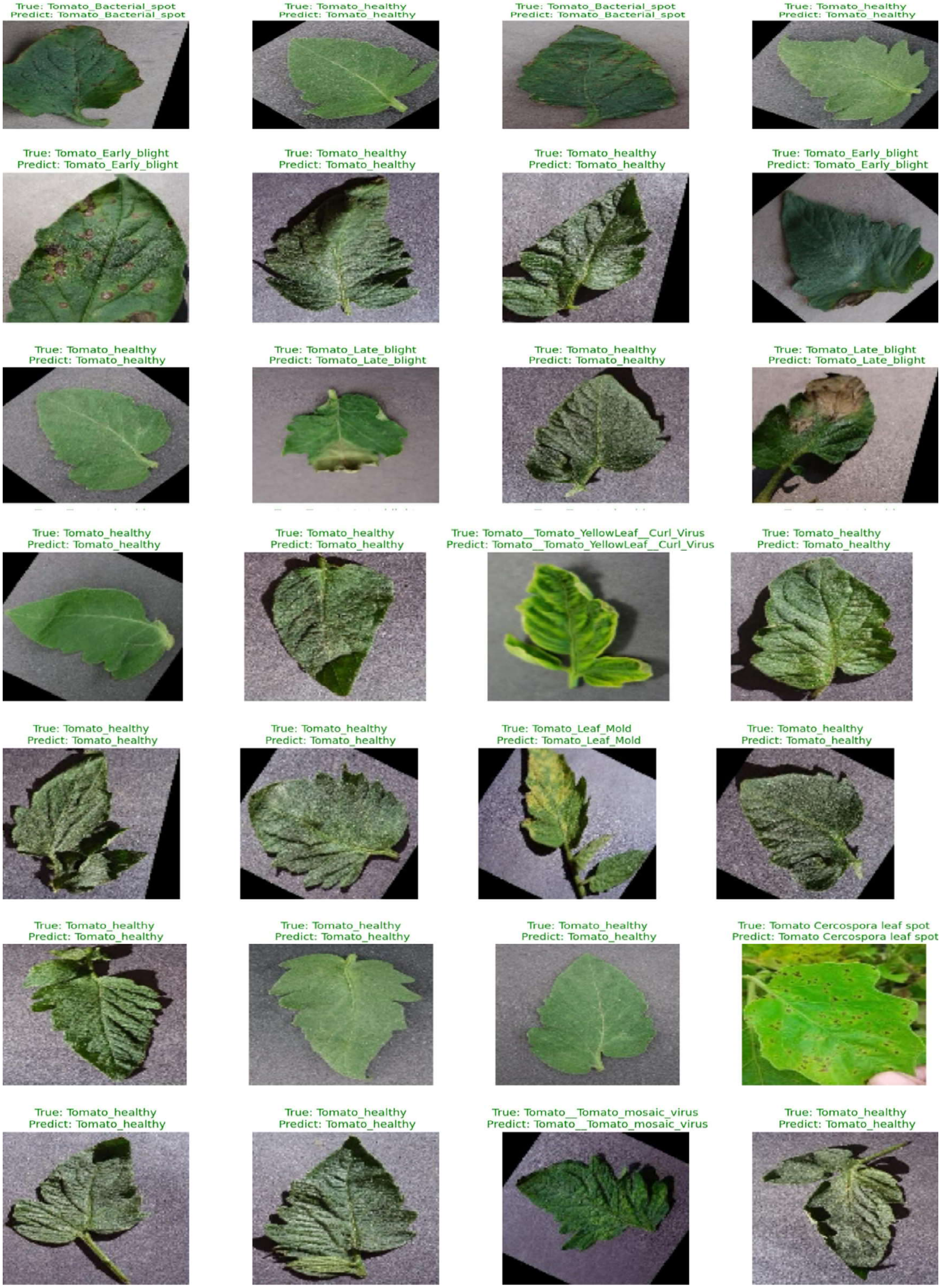
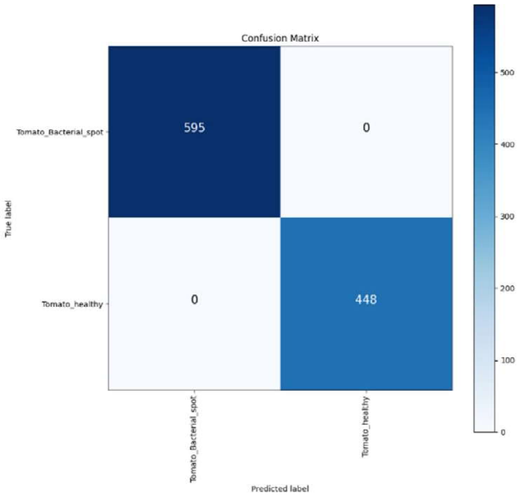
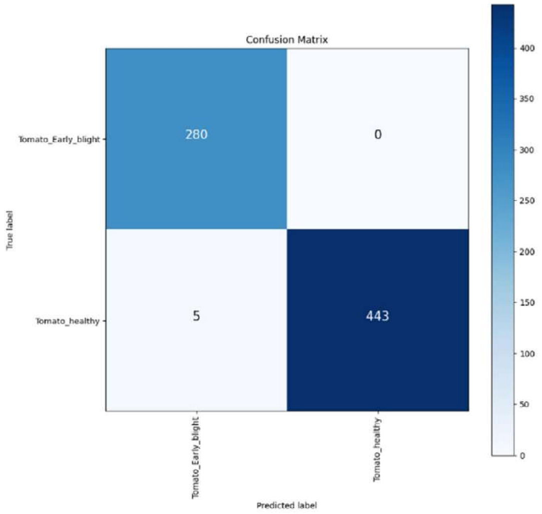


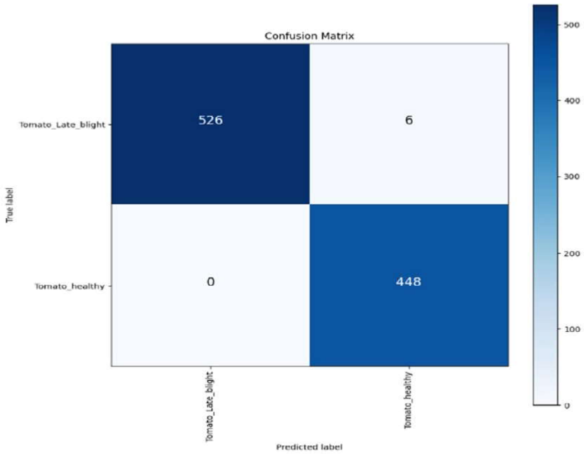
Fig. 5 : True and predicted results of diseased and healthy leaf images



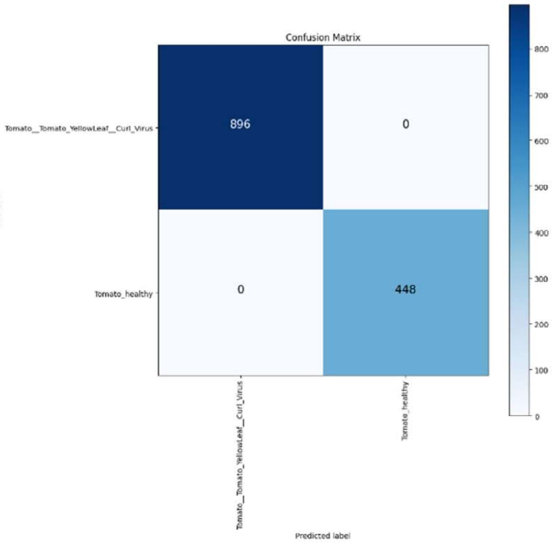
a) Bacterial spot



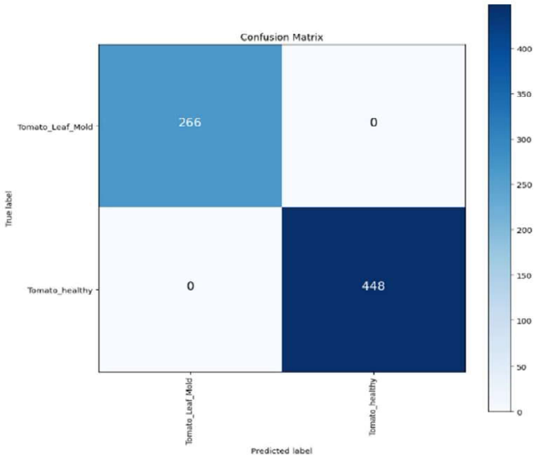
b) Early blight



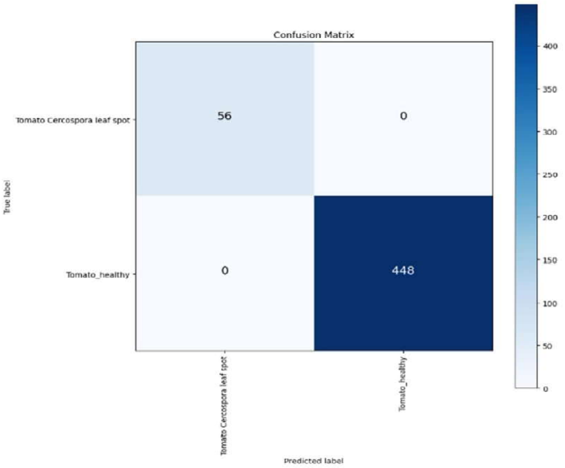
c) Late blight



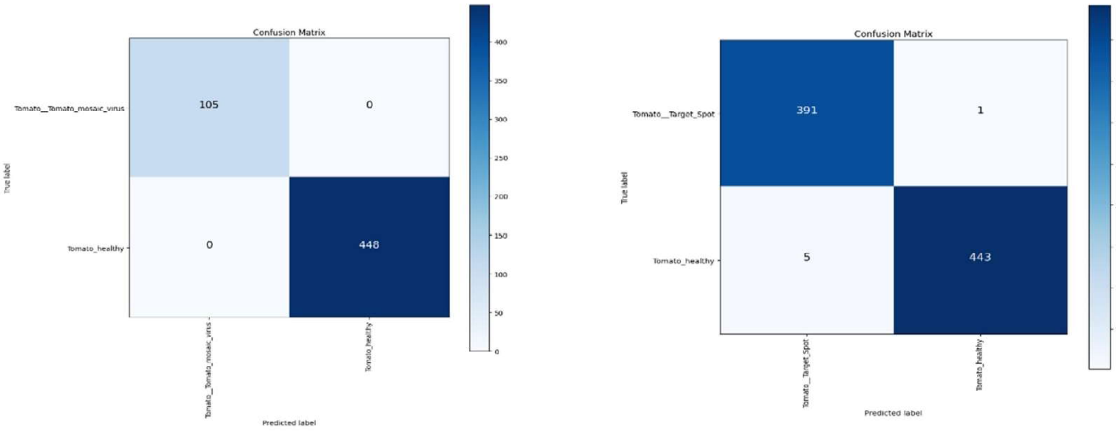
d) Leaf curl virus



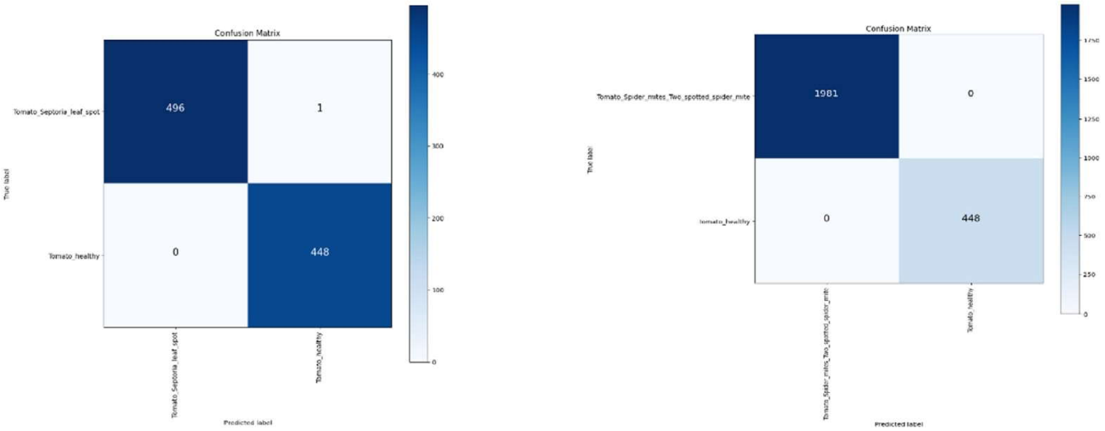
e) Leaf mold



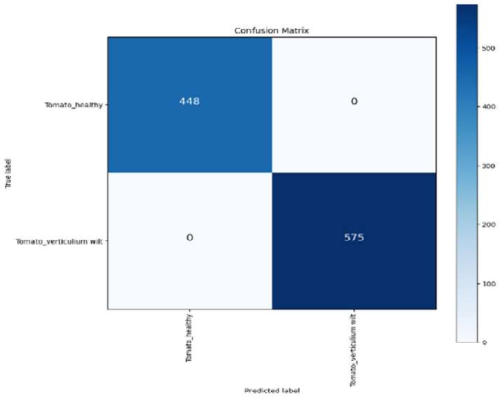
f) Cercospora leaf spot



g) Mosaic virus h) Target spot



i) Septoria leaf spot j) Two spotted mite



k) Verticillium wilt

Fig. 6 : Confusion matrix between tomato disease and healthy image dataset in custom CNN model

391 (98.73%), and 496 (99.79%) images, respectively. All healthy images were correctly identified in this study. In contrast, Brahimi et al. (2017) reported accuracy between 94.53 and 95.46 per cent of their CNN model in recognising 14828 images of yellow leaf curl virus, mosaic virus, target spot, spider mites, leaf mold, septoria spot, late blight, early blight, bacterial spot. Likewise, the model developed by Abouelmagd et al. (2024) showed a high classification rate for healthy images (99%) but lower accuracy for the 10 diseased classes (62-98%), *viz*., two-spotted spider mite, target spot, tomato mosaic virus, yellow

leaf curl virus, bacterial spot, early blight, late blight, leaf mould, septoria leaf spot diseased images.

CONCLUSION

The custom CNN model is expected to provide high accuracy and precision in distinguishing between healthy tomato plants and those afflicted with various diseases. Such reliable disease identification is crucial for timely intervention and mitigation strategies. As a continuous monitoring tool, this model enables the early detection of disease signs in tomato crops, facilitating proactive measures to prevent extensive outbreaks and reduce crop losses. This capability underscores the model's value in agricultural management and disease control.

REFERENCES

Abbas, A., Jain, S., Gour, M., & Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Computers and Electronics in Agriculture*, *187*, 106279. https://doi.org/10.1016/j.compag.2021.106279

Abouelmagd, L. M., Shams, M. Y., Marie, H. S., & Hassanien, A. E. (2024). An optimized capsule neural networks for tomato leaf disease classification. *EURASIP Journal on Image and Video Processing*, *2*, 1-21. https://doi.org/10.1186/s13640-023-00618-9

Afifi, A., Alhumam, A., & Abdelwahab, A. (2020). Convolutional neural network for automatic identification of plant diseases with limited data. *Plants*, *10*, 28. https://doi.org/10.3390/plants10010028

Afify, M., Loey, M., & Elsawy, A. (2022). A robust intelligent system for detecting tomato crop diseases using deep learning. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1-21. https:/ /doi.org/10.4018/IJSSCI.304439

Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). To LeD: Tomato leaf disease detection using convolution neural network, *Procedia Computer Science*, *167*, 293-301. doi: 10.1016/j.procs.2020.03.225

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. ArXiv, (Sept.), 1-33. arXiv ID: arXiv:1206.5533.

Bouni, M., Douzi, K., Hssina, B., & Douzi, S. (2023). Deep learning identifies tomato leaf disease by comparing four architectures using two types of optimizers. *Journal of Electrical and Computer Engineering*, 1-6. https://doi.org/10.1155/2023/5051005.

Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep learning for tomato diseases: classification and symptoms visualization. *Applied Artificial Intelligence*, *31*(4), 299–315. https://doi.org/10.1080/08839514.2017.1315516

Cagli, E., Dumas, C., & Prouff, E. (2017). Convolutional neural networks with data augmentation against jitter-based countermeasures. In: *International Conference on Cryptographic Hardware and Embedded Systems*, *CHES 2017*, *10529,* 45-68. https://doi.org/10.1007/978-3-319-66787-4_3

Choudhary, M. K., & Hiranwal, S. (2021). *Feature selection algorithms for plant leaf classification: a survey*. In: *Proceedings of international conference on communication and computational technologies*, edited by S. D. Purohit, D. Singh Jat, R. C. Poonia, S. Kumar, and S. Hiranwal (Springer, Singapore, 2021), 657–669. doi: 10.1007/978-981-15-5077-5_59

Emebo, O., Fori, B., Victor, G., & Zannu, T. (2019). Development of tomato septoria leaf spot and tomato mosaic diseases detection device using raspberry Pi and deep convolutional neural networks. *Journal of Physics*. Conference Series. https://doi.org/10.1088/1742-6596/1299/1/012118

Foysal, M., Ferdouse, A., Islam, M. S., Abujar, S., & Hossain, S. A. (2020). A novel approach for tomato diseases classification based on deep convolutional neural networks. *Proceedings of International Joint Conference on Computational Intelligence*, 583–91. https://doi.org/10.1007/978-981-13-7564-4_49

Gulati, A., Singh, A. P., & Chug, A. (2024). Disease recognition and prediction in tomato using machine learning algorithms. In: *Proceedings of the National Seminar on Opportunities & Challenges in Agriculture*, *Environmental & Biological Sciences for Global Sustainable Development*, organized by the Department of Botany, Zoology & Chemistry, Gandhi

Memorial National College, Ambala Cantt, Haryana. (9.4.2024): AEBS-37: 48 pp.

Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. ICT Express, *6*, 312–315. https://doi.org/10.1016/j.icte.2020.04.010

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. In: *Proceedings of IEEE*, *86*(11), 2278–2324. doi: 10.1109/5.726791

Masters, D., & Luschi, C. (2018). Revisiting small batch training for deep neural networks. *ArXiv*: 1-18. https://doi.org/10.48550/arXiv.1804.07612

Mikołajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. In: *International Interdisciplinary PhD Workshop*, *IIPhDW*. 117-122. doi: 10.1109/IIPHDW.2018.8388338

Poojary, R. P., Raina, R., & Mondal, A. K. (2021). Effect of data-augmentation on fine-tuned CNN model performance. *IAES International Journal of Artificial Intelligence* (IJ-AI), *10*(1), 84~92. https://doi.org/10.11591/ijai.v10.i1.pp84-92

Radiuk, P. (2017). Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. *Information Technology and Management Science*, *20,* 20-24. vrs: *it mac*:*v20*:y:2017:*is*:1:p:20-24:n:3

Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia Computer Science*, *133*, 1040–1047. https://doi.org/10.1016/j.procs.2018.07.070

Saini, M., & Seba, S. (2019). Comparison of deep learning, data augmentation and bag of-visual-words for classification of imbalanced image datasets. In: *Communications in Computer and Information Science book series CCIS, 1035,* 561-571. doi: 10.1007/978-981-13-9181-1_49

Sardogan, M., Tuncer, A., & Ozen, Y. (2018). Plant leaf disease detection and classification based on CNN with LVQ algorithm. In: *Proceedings of 3rd international conference on computer science and engineering*, 382–385 pp. doi: 10.1109/UBMK.2018.8566635

Shoaib, M., Shah, B., EI-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Hussain, T., & Ali, F. (2023). An advanced deep learning models-based plant disease detection: A review of recent research. *Frontiers in Plant Science*, 1-22. https://doi.org/10.3389/fpls.2023.1158933

Shorten, C., & Khoshgoftaar, T. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(60), 1-48. https://doi.org/10.1186/s40537-019-0197-0

Shruthi, U., Nagaveni, V. Arvind, C. S., & Sunil, G. L. (2022). *Tomato plant disease classification using deep learning architectures: a review*. Proceedings of second international conference on advances in computer engineering and communication systems: ICACECS 2021, 153–169. doi: 10.1007/978-981-16-7389-4_15

Sonika, Gulati, R., & Jangra, M. (2017). Incidence of *Tetranychus urticae* Koch on brinjal under field and screen house conditions. *Emergent Life Sciences Research*, *3*(2), 16-22. http://dx.doi.org/10.7324/ELSR.2017.321622

Vinayedula, (2023). How does epoch affect accuracy in deep learning model? In: https://www.geeksforgeeks.org/how-does-epoch-affect-accuracy-in-deep-learning-model/

Widiyanto, S., Fitrianto, R., & Wardani, D. T. (2019). Implementation of convolutional neural network method for classification of diseases in tomato leaves. In: *Proceedings of 2019 4th International Conference on Informatics and Computing*. https://doi.org/10.1109/ICIC47613.2019.8985909

Zhang, K., Wu, Q., Liu, A., & Meng, X. (2018). Can deep learning identify tomato leaf disease? *Advances in Multimedia*, 1-10. https://doi.org/10.1155/2018/6710865

(Received : 5.8.2024; Revised : 10.9.2025; Accepted : 16.9.2025)