

Tomato Disease Detection from Tomato Leaf Images Using CNN-Based Feature Extraction, Feature Selection with Whale Optimization Algorithm, and SVM Classifier

Le Thi Thu Hong(✉), Nguyen Sinh Huy, and Doan Quang Tu

Institute of Information Technology, AMST, Ha Noi, Viet Nam
lethithuhong1302@gmail.com

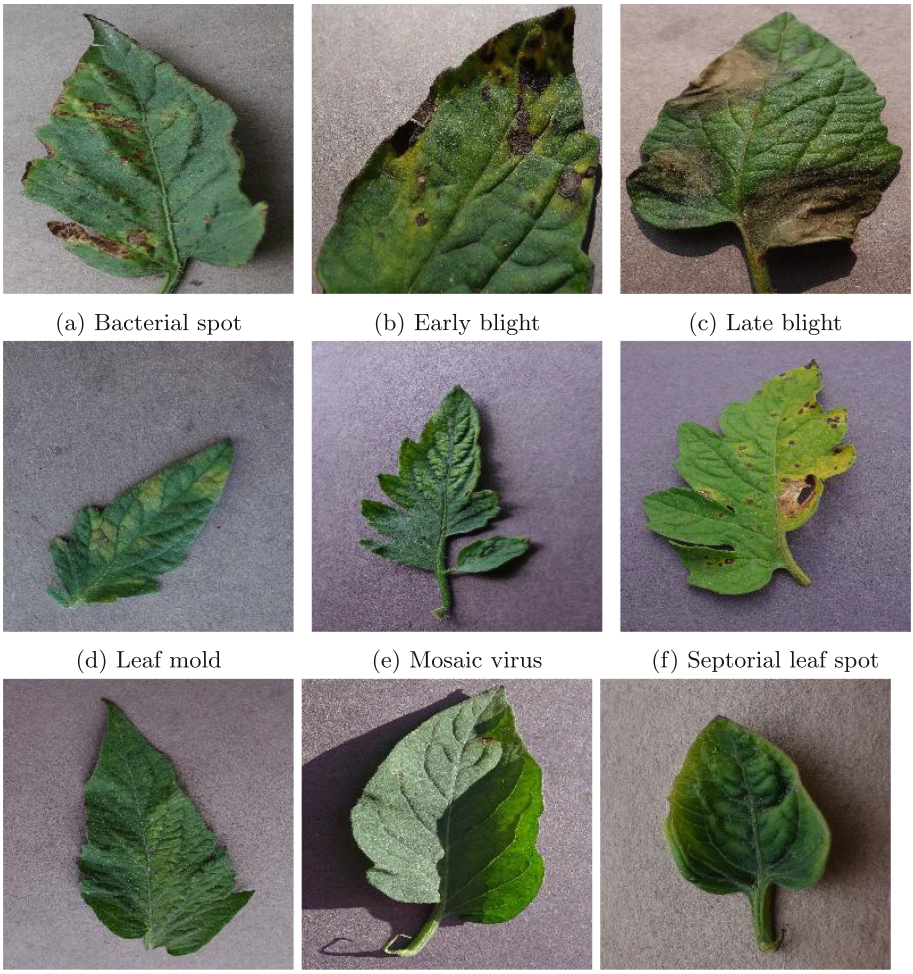
Abstract. Tomatoes, a popular crop, thrive on nearly any soil with good drainage. Identifying tomato diseases early on is crucial to maintaining quality and yield. This article proposes a hybrid model approach for early tomato disease detection from tomato leaf images. With the applied hybrid model approach, feature extraction was accomplished by employing a lightweight CNN model on images. These extracted features were then optimized by Whale optimization algorithm and utilized to train the SVM classifier for image classification. Validation of the test set utilizing Accuracy, Recall, Precision, and F1 score showed the proposed method achieved an accuracy score of 98.46%. Furthermore, the accuracy improvement obtained was 3.53% when using the proposed compared to the base CNN model. The findings validated that the proposed method outperforms state-of-the-art tomato disease detection methods with high accuracy and a lightweight model.

Keywords: Tomato Disease Detection · CNN-based Feature Extraction · Feature Optimization · SVM Classifier

1 Introduction

In agriculture, it is essential to detect plant diseases early to enhance productivity and the quality of agricultural products. Tomato plants stand out as renowned and vital crops due to their considerable market demand and nutritional benefits. However, insects and pests that prey on these plants and introduce various diseases hold the capacity to hinder the production of this popular commodity [5]. Diagnosing tomato plants at an early stage can save farmers from costly crop sprays and contribute to an increase in food production. Observing tomato plants closely for manual disease diagnosis is laborious and challenging. Farmers frequently encounter challenges in seeking advice from experts in distant areas and implementing preventive measures for uncommon diseases. Therefore, designing computer-aided systems for tomato crop disease recognition is critical.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024, LNCS 14813, pp. 192–205, 2024.
https://doi.org/10.1007/978-3-031-64605-8_14



(a) Bacterial spot (b) Early blight (c) Late blight

(d) Leaf mold (e) Mosaic virus (f) Septorial leaf spot

(g) Spider mite (h) Target spot (i) Yellow leaf curl virus

Fig. 1. Instances of tomato plant leaves affected by diseases

These systems can aid farmers in identifying diseased tomato plants, determining the specific diseases they are afflicted by, and utilizing that information to enhance the overall efficiency of the crop, minimizing losses.

The image classification models can be used to detect diseases in tomato plants from tomato leaf images by categorizing images of tomato leaves into healthy and diseased groups. The objective of this research is to detect and classify nine diseases commonly occurring in tomato plants, namely Early Blight, Late Blight, Septoria Leaf Spot, Yellow Leaf Curl Virus, Bacterial Spot, Mosaic virus, Target Spot, Spider Mite and Leaf Mold. Figure 1 is an example of tomato

plant leaves affected by disease, taken from the PlantVillage dataset [19]. We proposes a hybrid model approach for tomato disease detection based on tomato leaf images. Firstly, features were extracted from the leaf images using a lightweight CNN; then, to select the optimal feature subset, we applied feature optimization using the Whale optimization algorithm [10]. Finally, an SVM (Support vector machines) [7] classifier automatically categorizes data into one of ten classes: healthy or diseases. The proposed model is lightweight with a small parameter count and minimal memory requirements. The main contribution of this study includes:

1) Implementing a lightweight CNN to extract features from tomato leaf images for classifying them into ten classes, including the healthy and nine diseases.
2) Performing feature selection using a Whale optimization algorithm to improve tomato disease detection accuracy.
3) Utilizing an SVM classifier for tomato leaf image classification.
4) Using public open-access PlanVillage dataset for training and testing the proposed model. The results on the testing set, separate from the training set, indicate that the model achieves high accuracy with 98.46%.

The remainder of this article consists of "Related Work", "Proposed Method", "Experiments and Results" "Discussions" and "Conclusions" sections. Some existing research related to the same topic is included in the "Related Work" section. The "Proposed Method" section explains feature extractors based on the CNN models, feature optimization using the Whale optimization algorithm, and SVM classifier in detail. The "Experiments and Results" section includes all calculation and classification results from the proposed method. Finally, conclusions are drawn in the "Conclusions" section.

2 Related Work

Many proposed methods for predicting crop diseases have been based on machine learning in recent years. The suggested methodologies can be categorized into two primary strategies: conventional machine learning methods employing manually engineered feature extraction techniques, and deep learning techniques that directly analyze images using CNNs to identify crop diseases. Traditional machine-learning techniques often rely on segmenting crop images and subsequently classifying diseases based on features extracted from the segmented images. Zhang, Shanwen, et al. [19] proposed using image segmentation, PHOG features, and classification rules to identify diseases in apple and cucumber plants. In [17] the authors focused on detecting plant diseases. Utilizing the KNN classifier, they achieved a commendable accuracy of 98.56% when predicting plant leaf diseases. Mohammed A. Hussein and Amel H. Abbas [6] proposed an automatic disease detection system for tomato plants, wheat, and cucumbers. They employ preprocessing methods like cropping, resizing, fuzzy histogram equalization, and extraction of color and texture features. Then, they utilize the results to train a support vector machine classifier to detect and diagnose

plant leaf diseases. In [12], the image segmentation employed the GrabCut algorithm, and sample features were extracted using the LBP method. Subsequently, the SVM was trained on the extracted key points to classify leaf abnormalities. The proposed approach effectively identifies crop diseases, achieving an accuracy of 95%.

Recently, research based on CNN has been conducted in agriculture, and CNN analysis has demonstrated high performance in plant disease detection. In a study by Maeda-Gutiérrez, V. et al. [8], various deep learning-based models such as AlexNet, GoogleNet, Inception V3, ResNet-18, and ResNet-50 were assessed for categorizing tomato plant leaf samples into different classes. The GoogleNet model yielded the best results, attaining an accuracy of 99.39%. Zhao et al. [20] introduced a method for classifying various diseases in tomato plant leaves, employing a CNN with an integrated attention mechanism. This approach achieved an accuracy rate of 99.24%. Bhujel et al. [3] introduced a deep learning-based method, specifically employing the ResNet18 model with CBAM, for classifying diseases in tomato plant leaves. This approach achieved an impressive accuracy of 99.69%. Richey et al. [14] suggested a method for detecting maize crop diseases with an accuracy of 99%, employing the ResNet50 framework for feature computation and class determination. Although these methods yield very high results, they are not suitable for mobile device applications with limited resources due to high processing requirements.

This article proposes a hybrid model approach for tomato disease detection from tomato leaf images with feature extraction accomplished by employing a CNN model on images. These extracted features were then optimized and utilized to train the SVM classifier. The proposed method achieves high performance with a lightweight model. It can be a good choice for human expert-level crop disease diagnostics mobile systems.

3 Proposed Method

The objective of this study is to tackle the issue of classifying tomato leaf images by predicting the categories of tomato plant diseases (Healthy and nine specific diseases) based on their corresponding leaf images. We proposed a hybrid model approach, shown in Fig. 2, to achieve this objective. First, the tomato leaf images are collected from the PlantVillage database. Then, the images are resized and augmented before passing through the base CNN. Next, the feature from images is extracted using CNN. The features obtained from CNN extractors are optimized using the Feature Selection based on Whale Optimization Algorithm (FSWOA) to get a dominant feature vector, which is input to SVM classifier for final classification.

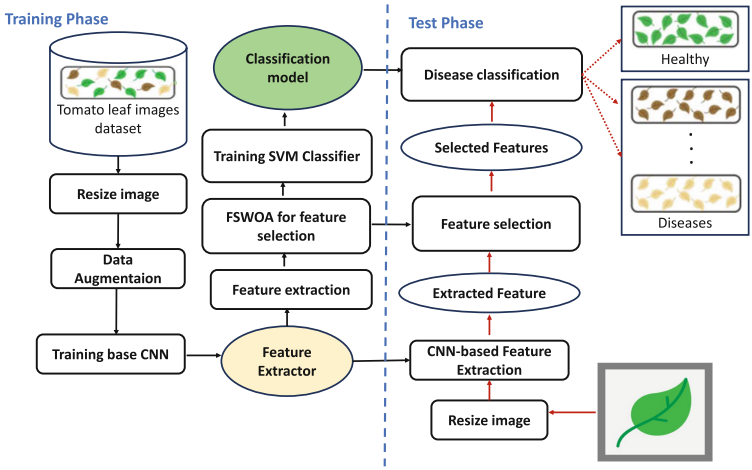


Fig. 2. The proposed method for Tomato disease detection

3.1 Dataset and Preprocessing

This study used tomato leaf images from PlanVillage, an open-access repository of plant leaf images for deploying plant disease detection models, to train and test the proposed model. The PlantVillage dataset comprises a total of 54,306 leaf images representing 14 types of plants. In this study, we only utilize images of tomato plant leaves from the PlantVillage dataset, belonging to 10 classes. Table 1 provides statistics on the number of tomato leaf images belonging to different categories, and Fig. 1 is an example of them from the PlantVillage dataset.

Table 1. The number of tomato leaf images in PlantVillage dataset.

No.	Category name	Total	No.	Category name	Total
1.	Healthy	1591	6.	Yellow Leaf	5357
2.	Bacterial spot	2127	7.	Septoria leaf spot	1771
3.	Early blight	1000	8.	Target spot	1404
4.	Late blight	1909	9.	Mosaic virus	373
5.	Two spotted spider mite	1676	10.	Leaf mold	952

The PlanVillage dataset includes RBG leaf image files with a dimension of 256×256x3 pixels. Thus, it is necessary to resize images to match the input requirements of the CNN extractors. Using Open-CV, We resized the collected images to $224 \times 224 \times 3$ pixels. Furthermore, we have employed data augmentation to reduce the over-fitting problem in training models. The techniques

employed for augmentation include flipping vertically and horizontally, rotating randomly within a range of –10 to 10 °C, scaling randomly between 0.5 and 1.5, shearing randomly within –5 to 5 °C, applying random Gaussian blur with a sigma of 3.0, normalizing contrast randomly by a factor of 1 to 1.5, adjusting brightness randomly within a range of 1 to 1.5, and performing random cropping and padding by 0–5% of both height and width.

3.2 The CNN-Based Feature Extractor

Feature extraction plays a vital role in enhancing the efficiency of the process and generating more meaningful datasets with larger and higher-quality images. In contrast to alternative feature extraction techniques, CNNs can directly capture the images' features within the input dataset. CNN architecture comprises convolutional layers (CLs), pooling layers (PLs), fully connected layers, and rectified linear units (ReLU). CLs specialize in learning convolutions and optimizing data categorization performance. Pooling layers PLs are vital in mitigating overfitting, ensuring stable transformation, and improving computational efficiency by reducing the convolution's structural output. The ReLU activation function enhances the network's non-linear characteristics [2]. When CNNs are utilized as feature extractors, the features are derived from the convolutional and fully connected layers of the CNNs, and these features encompass abstract visual attributes.

In this study, we implemented a simple CNN (called base CNN) for feature extraction to build a compact model suitable for mobile applications. Details of the base CNN layers are presented in Table 2. The base CNN is relatively small, with six pairs of Convolution and MaxPooling layers, and the total number

Table 2. Summary of the base CNN for Feature Extractor.

Layer	Output Shape	Param#
input (InputLayer)	[(224, 224, 3)]	0
conv2d-1(Conv2D-3x3 Kernel,Depth 32)	(222, 222, 32)	896
maxpooling2d-1 (MaxPooling2D-2x2)	(111, 111, 32)	0
conv2d-2 (Conv2D-3x3 Kernel, Depth 64)	(109, 109, 64)	18496
maxpooling2d2 (MaxPooling2D-2x2)	(54, 54, 64)	0
conv2d-3 (Conv2D- 3x3 Kernel, Depth 64)	(52, 52, 64)	36928
maxpooling2d-3 (MaxPooling2D- 2x2)	(26, 26, 64)	0
conv2d-4 (Conv2D- 3x3 Kernel, Depth 64)	(24, 24, 64)	36928
maxpooling2d-4 (MaxPooling2D- 2x2)	(12, 12, 64)	0
conv2d-5 (Conv2D- 3x3 Kernel, Depth 64)	(10, 10, 64)	36928
maxpooling2d-5 (MaxPooling2D- 2x2)	(5, 5, 64)	0
conv2d-6 (Conv2D-3x3 Kernel, Depth 64)	(3, 3, 64)	36928
max-pooling2d-6 (MaxPooling2D- 2x2)	(1, 1, 64)	0
flatten (Flatten)	(64)	0
dense (Dense with SoftMax activation)	(10)	650
Total params: 167,754		

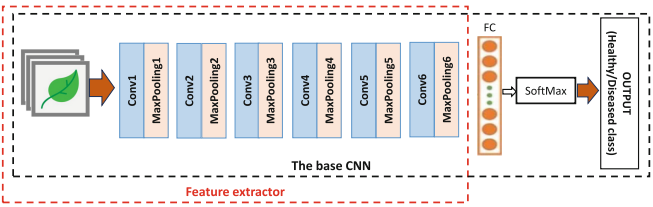


Fig. 3. The architecture of the Feature Extractor

of trainable parameters is only 167,754. Figure 3 illustrates the architecture of the Feature Extractor. First, the base CNN model was instantiated with layers presented in Table 2, and then it was trained from scratch using tomato leaf images from the PlantVillage dataset. Finally, the Feature extractor was obtained by removing the last layer (the Fully Connected layer with SoftMax activation).

3.3 Feature Selection with Whale Optimization Algorithm

In the proposed scheme, after CNN feature extractors, we receive feature vectors with a length of 64. Not all of these features are useful for fitting the classification models. Using unnecessary features diminishes the model's generalization ability and potentially lowers the classifier's accuracy. Furthermore, adding the number of features in a model elevates its overall complexity. Thus, it is necessary to have a feature selection procedure in the proposed scheme to avoid the over-fitting problem and improve the model's performance.

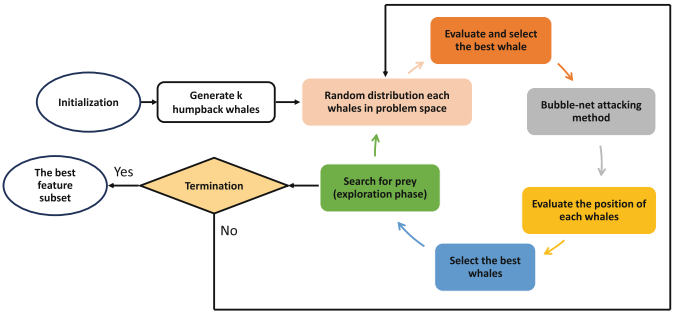


Fig. 4. Feature Selection based on Whale Optimization Algorithm

The conventional approaches for feature selection utilize a selected mathematical framework to decrease the feature vector, whereas heuristic algorithm-based (HA-based) feature optimization is commonly used to diminish the dimension of the feature vector. The Whale optimization algorithm, devised by Mirjalili

and Lewis [9], is an effective heuristic algorithm that emulates humpback whales' foraging and hunting strategies. In this study, we use the FSWOA (Feature Selection based on Whale Optimization Algorithm), proposed by Zamani et al. [18] for feature selection. Figure 4 depicts the flowchart of FSWOA. The algorithm comprises three primary phases: initially, it generates k whales and disperses them randomly throughout the search space. Subsequently, each whale's position undergoes evaluation to identify the optimal whale; the remaining whales then adjust their positions relative to the best whale's position. In the second phase, whales engage in a bubble-net attack, akin to an exploitation stage, where each whale proposes a feature subset. These subsets are evaluated based on the classifier's accuracy on the testing set, with the highest accuracy determining the best whale. Finally, in the third phase, or the exploration stage, whales pursue prey randomly based on their respective positions.

3.4 The SVM Classifier

The last part of the proposed scheme is a classifier that automatically categorizes data into one of ten classes. In this work, the SVM classifier was implemented and evaluated for tomato leaf disease classification. Support Vector Machine (SVM) [7] is a class of supervised learning algorithms for classification and regression tasks. They are extensively utilized due to their capability in handling complex datasets, as they identify the optimal decision boundary, also known as a hyperplane, that maximizes the margin between distinct classes. The mathematics behind this process is described as follows:

$$W * x + b = 0 \tag{1}$$

SVMs can be primarily categorized into two types: linear SVM and non-linear SVM. In the case of linear SVM, the algorithm seeks to identify an optimal hyperplane that effectively separates two classes in a dataset exhibiting linear separability. The margin is calculated as the distance between the hyperplane and the closest data points from each class. However, it's important to note that numerous real-world datasets are not linearly separable. Non-linear SVM expands the algorithm's ability to address such datasets by mapping them into a higher-dimensional feature space. This transformation is accomplished using kernel functions like Polynomial, Radial Basis Function (RBF), and Sigmoid Function, which enables SVM to identify non-linear decision boundaries. In this work, we use a non-linear SVM classifier with RBF kernel.

4 Experiments and Results

4.1 Experimental Setting

We use a dataset consisting of 18,136 tomato leaf images belonging to 10 classes from the PlantVillage dataset for training and testing the model. The training set consists of 16,344 images, and the test set consists of 1,792 images. The

data was divided into training and validation sets in the training phase, with 13075(80%) allocated to training and 3269(20%) to validation. We train the base CNN to classify 10 classes of tomato leaves from scratch. The entire network was updated via the Adam optimizer, and Adam's learning rate was set to 0.0001. The model produced during the epoch with the highest accuracy value on the validation set is our final model. All algorithms were developed and trained using the Keras framework with a Tensorflow backend and the Scikit learn library on a PC with a GeForce GTX 1080 Ti GPU.

In this study, to attain a thorough assessment of a model's performance in classification task, we employ well-known evaluation functions such as, Accuracy, Recall, Precision, and F1-score. These metrics are utilized to evaluate the congruence between the model's classification results and the assigned class labels for the tomato leaf images. The Recall, as depicted in Eq. (2), signifies the correct classification of samples within the positive classes. Precision, also known as Positive Predictive Value (PPV), is determined by Eq. (3). The F1 score amalgamates both precision and recall into a unified metric, calculated using Eq. (4). The proposed algorithm's cost function is defined by the classifier's accuracy, as illustrated in Eq. (5), where the sum of True Positives (TP) and False Positives (FP) represents the total number of subjects with positive tests, and the sum of False Negatives (FN) and True Negatives (TN) represents the total number of subjects with negative tests.

$$Recall = \frac{TP}{(TP + FN)} \tag{2}$$

$$Precision(PPV) = \frac{TP}{(TP + FP)} \tag{3}$$

$$F1 = 2 \times (\frac{precision \times recall}{(precision + recall)}) \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

4.2 Performance of the Base CNN for Tomato Disease Detection

In the initial step, we trained and tested the base CNN for tomato disease image classification. The network employ the Softmax classifier to perform the classification task. The Fig. 5 illustrates the accuracy and loss curves of the CNN throughout the training process. As described in Fig. 5, the training and validation sets' accuracy approximates each other and gradually increases towards 1. Meanwhile, the loss values on the training and validation sets exhibit a closely correlated relationship and tend to decrease towards the value of 0. This indicates that the classification task leads to convergence for the network.

Table 3 shows the accuracy results achieved by the base CNN on the test set. On average, across all classes, the CNN achieved an accuracy of 94.93%, with precision and recall values both equal to 95%. The highest accuracy was

attained in the 'Healthy' and 'Yellow Leaf Curl Virus' classes, reaching 100%. Meanwhile, the 'Two-spotted spider mite' class had the lowest accuracy, reaching only 86%. The average accuracy of the model is acceptable; however, there is a significant disparity in accuracy across different classes. This motivates us to continue improving the model to enhance accuracy and prediction balance across various classes.

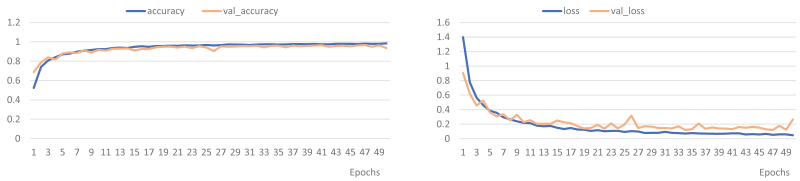


Fig. 5. The CNN base Training Process

Table 3. Accuracy of the base CNN on test set.

Category name	Samples#	Acc(%)	PPV(%)	Recall(%)	F1-score(%)
Bacterial spot	210	95.24	97	95	96
Early blight	110	89.09	92	89	90
Late blight	179	97.77	95	98	96
Leaf Mold	88	95.45	99	95	97
Septoria leaf spot	195	90.26	96	90	93
Two-spotted spider mite	167	86.23	94	86	90
Target Spot	122	90.16	85	90	87
Yellow Leaf Curl Virus	544	98.71	100	99	99
Tomato mosaic virus	39	97.44	100	97	99
Healthy	162	100.00	85	100	92
The average	*1816*	*94.93*	*95*	*95*	*95*

4.3 Tomato Disease Detection Performance of SVM Classifier with FSWOA

After training the base CNN, we use part of the CNN as feature extractors to extract features of tomato leaf images. These extracted features were then optimized using the FSWOA and utilized to train the SVM classifier. Finally, we used the trained SVM classifier to classify the tomato leaf images in the test set. Table 4 shows the accuracy results achieved by the proposed model. The average accuracy of the proposed model across all classes is quite high, reaching 98.53%. Absolute accuracy was achieved 100% in the 'Healthy' and

'Tomato mosaic virus' classes. Additionally, the accuracy in the remaining classes is consistently high, ranging from 97% to 99%. These results surpass the base CNN model significantly. Table 5 compares the accuracy between the proposed model and the base CNN model, while Fig. 6 compares the confusion matrices of the two models. Figure 6 shows that the number of correct predictions for each class in the proposed model has increased compared to the base CNN model. The proposed model shows an average accuracy improvement of 3.53% compared to the base CNN model. There is an improvement in accuracy across all classes, with the most considerable improvement observed in the 'Two-spotted spider mite' class at 10.69%. Furthermore, the accuracy across classes in the proposed model is more uniform compared to the base CNN model.

Table 4. Accuracy of SVM classifier with FSWOA on test set.

Category name	Samples#	Acc(%)	PPV(%)	Recall(%)	F1-score(%)
Bacterial spot	210	97.62	99	98	98
Early blight	110	97.27	91	97	94
Late blight	179	99.44	99	99	99
Leaf Mold	88	96.59	99	97	98
Septoria leaf spot	195	96.59	99	97	98
Two-spotted spider mite	167	96.92	97	100	98
Target Spot	122	96.72	96	97	96
Yellow Leaf Curl Virus	544	98.90	100	99	99
Tomato mosaic virus	39	100.00	100	100	100
Healthy	162	100.00	100	100	100
The average	*1816*	*98.46*	*98.46*	*98.46*	*98.46*

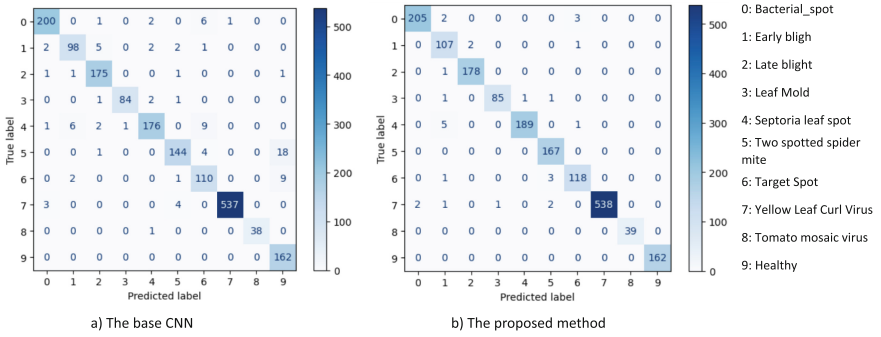


Fig. 6. Comparation confusion matrixes of the base CNN and proposed method

Table 5. Improvement of the proposed method compares to the base CNN.

Category name	Accuracy of base CNN	Accuracy of proposed method	Improvement
Bacterial spot	95.24	97.62	2.38
Early blight	89.09	97.27	8.18
Late blight	97.77	99.44	1.67
Leaf Mold	95.45	96.59	1.14
Septoria leaf spot	90.26	96.59	6.33
Two-spotted spider mite	86.23	96.92	10.69
Target Spot	90.16	96.72	6.56
Yellow Leaf Curl Virus	98.71	98.9	0.19
Tomato mosaic virus	397.44	100	2.56
Healthy	100	100	0
The average	*94.93*	*98.46*	*3.53*

4.4 Comparative Discussion

This section presents the tomato disease detection results from existing methods. We have chosen the studies that used the PlantVillage dataset the same way we did. Due to the absence of evaluation scenarios facilitating an equitable comparison between methods and the unavailability of code for the methods mentioned, the comparison was conducted based on the performance metrics documented in the literature. The results are presented in Table 6. From Table 6, it can be observed that the accuracy of tomato disease detection in current studies is quite high. Some studies achieved approximately 100% accuracy, such as [4,11]. Our proposed model also gains a relatively high accuracy, ranking second among the mentioned models. Meanwhile, the model's parameter count is the smallest, approximately only 0.8% of the highest accuracy model [11], and our model's storage space is also minimal. This makes our model more compact, requiring fewer computational resources and achieving faster execution times. These advantages are crucial in the context of mobile systems supporting tomato disease detection using tomato leaf images.

Table 6. Comparing methods for tomato disease detection

Papers	Model	Acc	Storage Space	Params#
Rangarajan et al.(2018) [13]	VGG19	97.49%	n/a	143.7M
Agarwal et al. (2019) [1]	CNN	91.2%	1,696KB	208,802
Trivedi et al. (2021) [16]	CNN	98.49%	22,565KB	1,422,542
Nawaz et al. (2022) [11]	ResNet-34	**99.97%**	n/a	21.5M
Sakkarvarthi et al. (2022) [15]	CNN	88.17%	n/a	1,060,138
Debnath et al. (2023) [4]	EfficientNetV2B2	99.80%	n/a	10.2M
Proposed method	CNN Hybrid	98.46%	**409 KB**	**167,754**

5 Conclusion

This work proposes a hybrid model approach for tomato disease detection using tomato leaf images. Firstly, features were extracted from the images using a simple CNN. Then, the extracted features were optimized using the FSWOA algorithm. Finally, an SVM classifier automatically categorizes data into one of ten classes: one healthy class and nine disease classes. The proposed model was trained and tested using the PlantVilliage dataset, a publicly available dataset of plant leaf images. The test results on an independent test set, separate from the training set, indicate that the model achieves a high accuracy of 98.46%. Furthermore, the proposed model is lightweight, with a small parameter count and minimal memory requirements. Therefore, the proposed method can be utilized for mobile systems. The main limitation of this study is that we tested the proposed method on only one publicly available dataset of tomato leaf images. Therefore, to affirm the generalizability of the proposed method, further testing will be conducted on different datasets, not limited to tomato leaf images but potentially including many other plant leaves.

References

1. Agarwal, M., Singh, A., Arjaria, S., Sinha, A., Gupta, S.: Toled: tomato leaf disease detection using convolution neural network. Procedia Comput. Sci. **167**, 293–301 (2020)
2. Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET), pp. 1–6. IEEE (2017)
3. Bhujel, A., Kim, N.E., Arulmozhi, E., Basak, J.K., Kim, H.T.: A lightweight attention-based convolutional neural networks for tomato leaf disease classification. Agriculture **12**(2), 228 (2022)
4. Debnath, A., et al.: A smartphone-based detection system for tomato leaf disease using efficientnetv2b2 and its explainability with artificial intelligence (ai). Sensors **23**(21), 8685 (2023)
5. Elnaggar, S., Mohamed, A.M., Bakeer, A., Osman, T.A.: Current status of bacterial wilt (ralstonia solanacearum) disease in major tomato (solanum lycopersicum l.) growing areas in egypt. Arch. Agric. Environ. Sci **3**(4), 399–406 (2018)
6. Hussein, M.A., Abbas, A.H.: Plant leaf disease detection using support vector machine. Al-Mustansiriyah J. Sci. **30**(1), 105–110 (2019)
7. Ma, Y., Guo, G. (eds.): Support Vector Machines Applications. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-02300-7
8. Maeda-Gutiérrez, V., et al.: Comparison of convolutional neural network architectures for classification of tomato plant diseases. Appl. Sci. **10**(4), 1245 (2020)
9. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Adv. Eng. Softw. **95**, 51–67 (2016)
10. Mirjalili, S., Mirjalili, S.M., Saremi, S., Mirjalili, S.: Whale optimization algorithm: theory, literature review, and application in designing photonic crystal filters. In: Mirjalili, S., Song Dong, J., Lewis, A. (eds.) Nature-Inspired Optimizers. SCI, vol. 811, pp. 219–238. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-12127-3_13

11. Nawaz, M., et al.: A robust deep learning approach for tomato plant leaf disease localization and classification. Sci. Rep. **12**(1), 18568 (2022)
12. Pantazi, X.E., Moshou, D., Tamouridou, A.A.: Automated leaf disease detection in different crop species through image features analysis and one class classifiers. Comput. Electron. Agric. **156**, 96–104 (2019)
13. Rangarajan, A.K., Purushothaman, R., Ramesh, A.: Tomato crop disease classification using pre-trained deep learning algorithm. Procedia Comput. Sci. **133**, 1040–1047 (2018)
14. Richey, B., Majumder, S., Shirvaikar, M., Kehtarnavaz, N.: Real-time detection of maize crop disease via a deep learning-based smartphone app. In: Real-Time Image Processing and Deep Learning 2020, vol. 11401, pp. 23–29. SPIE (2020)
15. Sakkarvarthi, G., Sathianesan, G.W., Murugan, V.S., Reddy, A.J., Jayagopal, P., Elsisi, M.: Detection and classification of tomato crop disease using convolutional neural network. Electronics **11**(21), 3618 (2022)
16. Trivedi, N.K., et al.: Early detection and classification of tomato leaf disease using high-performance deep neural network. Sensors **21**(23), 7987 (2021)
17. Tulshan, A.S., Raul, N.: Plant leaf disease detection using machine learning. In: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–6. IEEE (2019)
18. Zamani, H., Nadimi-Shahraki, M.H.: Feature selection based on whale optimization algorithm for diseases diagnosis. Int. J. Comput. Sci. Inf. Secur. **14**(9), 1243 (2016)
19. Zhang, S., Wang, H., Huang, W., You, Z.: Plant diseased leaf segmentation and recognition by fusion of superpixel, k-means and phog. Optik **157**, 866–872 (2018)
20. Zhao, S., Peng, Y., Liu, J., Wu, S.: Tomato leaf disease diagnosis based on improved convolution neural network by attention module. Agriculture **11**(7), 651 (2021)