Wiley
Journal of Engineering
Volume 2024, Article ID 9918296, 10 pages
https://doi.org/10.1155/2024/9918296

WILEY

Research Article

Automated Tomato Leaf Disease Detection Using Image Processing: An SVM-Based Approach with GLCM and SIFT Features

Rashid Khan,[1] **Nasir Ud Din**,[2] **Asim Zaman**,[3,4] **and Bingding Huang**[1]

[1]*College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China*
[2]*College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China*
[3]*College of Health Science and Environmental Engineering, Shenzhen Technology University, Shenzhen, China*
[4]*School of Biomedical Engineering, Shenzhen University Medical School, Shenzhen University, Shenzhen, China*

Correspondence should be addressed to Bingding Huang; huangbingding@sztu.edu.cn

Received 7 February 2024; Revised 26 May 2024; Accepted 13 August 2024

Academic Editor: Rosa De Finis

Tomato cultivation is increasingly widespread, yet it faces significant challenges, particularly from plant diseases caused by fungi, bacteria, and insects. Addressing these diseases is crucial for ensuring the quality and yield of tomato crops. To support specialists in accurately identifying and managing these diseases, we propose an advanced automatic system for detecting and identifying tomato leaf diseases using sophisticated image processing techniques. Therefore, we proposed an approach that employs robust feature extraction methods, including the gray level co-occurrence matrix (GLCM) and scale-invariant feature transform (SIFT), coupled with a support vector machine (SVM) for adequate classification. We curated an extensive dataset of 2700 tomato leaf images, with a minimum of 300 images for each of the nine distinct disease classes. This comprehensive dataset facilitated the training and testing of various machine learning and deep learning models. The experimental results highlight our proposed approach's exceptional accuracy and reliability, significantly improving the detection and classification of tomato leaf diseases. A thorough comparative analysis with contemporary state-of-the-art techniques further validates the superiority of our system. Our findings suggest that this framework can significantly benefit tomato cultivation by enabling timely and precise disease management. Future research can explore integrating advanced deep learning algorithms to enhance the system's accuracy in this multiclass classification challenge.

1. Introduction

Agriculture plays a vital role in the lives of people all around the globe. It is central to human life, impacting our food, medicine, clothing, and employment opportunities. The agricultural sector plays a crucial role in the economy of every country as it serves as the primary source of food production. Tomatoes hold a significant position among the various crops, accounting for approximately 16% of the market share [1]. Generally, the apparent increase in demand and production of tomatoes is indicated by the continuous flourishing of the international tomato market. In the era of 2005–2007, the market value increased at the +3.2% annual rate [2]. Over the examined period, some obvious variations were observed in the rapidly growing period. The market benefits increased by +8% in 2008 as compared to last year's level. In 2014, the worldwide tomato market achieved an extreme benefit of 200,014 million USD throughout the examined duration [2]. In 2015, the income of the international tomato market was up to 189,229 million USD, diminishing by −5.4% against the income of the last year [2]. The revenue of the total tomato market has become unsuccessful in recuperating its initial peak from 2015 to the end of the session under study. It happens because every year, tomato crops suffer many diseases, which results in a noticeable loss of yield [3]. The quality, yield, and quantity of tomatoes decrease because of improper care and diseases. Diseases affect crop yield and other crops in the vicinity [3].

Loss of thousands of billion dollars is estimated yearly because of disease attacks on crops [4]. It is hard to detect and control plant diseases at the right time to lessen crop loss and increase production. In this context, tomato stems, leaves, and fruit show disease symptoms, making them identifiable externally instead of the DNA study. In response to these challenges, our research introduces an innovative framework for automatically identifying tomato leaf diseases using visual symptoms appearing on leaves. Our approach leverages the power of image processing, with a focus on extracting valuable features from plant leaf images. These features, including gray-level co-occurrence matrix (GLCM) and scale-invariant feature transform (SIFT) features, are subsequently employed by a support vector machine (SVM) for accurate disease classification. The proposed framework promises to revolutionize how we manage tomato crops by enabling swift and precise disease detection, ultimately leading to enhanced crop productivity and quality. The significance of our research is underscored by its potential to address a pressing issue in agriculture while contributing to the field of intelligent systems and image processing. The rest of this paper is organized as follows. Section 2 briefly reviews the literature on the subject. Section 3 presents the proposed plant disease identification/classification framework. Section 4 reports the experimental results, and Section 5 concludes the paper.

2. Related Work

The main characteristic of our projected technique is that it is automatic, robust, and highly efficient for analyzing and detection purposes. A few years back, several automatic algorithms, such as simple and hybrid systems, had been developed for this purpose. Optical coherence microscopy (OCM) is a powerful method for analyzing constant and site-specific diseases using robust pattern-matching techniques [5]. The authors also validate its effectiveness for nonrigid plant object searching. Furthermore, we introduce a single-feature 2D xy-color histogram approach. This feature is input for disease classification and quantification in a support vector machine (SVM) classifier. The results demonstrate that the proposed method can be implemented in sugar beet fields for improved examination and quantification of foliar disease growth. In 2015, Santos et al. [6] proposed a machine vision system for diagnosing empathy in plants. The system focuses on analyzing visual signs of plant diseases through colored images. The crops' digital images were first segmented and enhanced to emphasize the unhealthy parts. Furthermore, a set of features was obtained from each image. The extracted features were subsequently used as input for the support vector machine (SVM) classifier. The evaluations were performed to determine the optimal classification model. It was hypothesized that a subset of more valuable features would exist within the realm of images based on their known characteristics. To test this assumption, multiple classification models were assessed using cross-validation. The study suggests that texture-related features could be useful in distinguishing targeted images that do not conform to distinct color or pattern shapes. In addition, machine vision systems could successfully discriminate objectives by providing suitable information. The classification [7] and automatic disease detection of plant leaves using an algorithm for the image segmentation technique were discussed in [8]. The survey on various techniques of disease classification that can be a source of disease detection of plant leaves is also covered. Image segmentation is an important characteristic for detecting diseases in plant leaves, completed with the use of the genetic algorithm. In 2014, Roberto et al. [9] discussed the multispectral imaging method that was applied to grapevine leaves containing 25 leaves for validation samples and 10 for calibration purposes. The images are captured with five view angles changing from 0° up to 75° at the camera's perpendicular position to the leaf's surface. The sensitivity of detection was assessed by relating to images for validation, and the algorithm was established on the grouping of 2 spectral indexes. The algorithm was individually proficient based on the images for the calibration. The complete results show that detection sensitivity usually increases with the increase in the view angle, with the highest value achieved for images attained at 60°. The algorithm's sensitivity offers dramatic improvement as follows: changes from 9% and 0% up to 73% and 60% for tissues with early-middle symptoms.

The various platforms for capturing hyperspectral images are discussed in [10]. The study also verifies that utilizing SVM and classification algorithms yields improved performance compared to other methods, such as linear and quadratic discriminant analysis and linear SVM. In 2016, Patil et al. [11] presented a content-based image retrieval (CBIR) system for analyzing soybean disease leaves. This system utilizes various features such as texture, color, and leaf shape. To extract color features, the system employs the HSV color histogram [12]. Shape features are typically represented by matching key points using the scale-invariant feature transform (SIFT) [13]. The Gabor filter and the local binary pattern (LBP) are commonly used texture characteristics. A novel texture characteristic named local gray Gabor pattern (LGGP) is proposed to enhance the texture analysis by combining LBP with Gabor. Three soybean leaf diseases are used to assess the retrieval precision of each feature. Additional elements such as color, shape, and texture are integrated to enhance performance. The findings show that combining LGGP, color histogram, and SIFT improves retrieval precision. When dealing with soybean leaves affected by a mosaic virus, Septoria brown spot, and pod mottle disease, they achieve retrieval efficiencies of approximately 96%, 68%, and 76%, respectively. Integrating characteristics yields average retrieval efficiencies of 80% for the top 5 retrievals and 72% for the top 5 retrievals. This retrieval precision is reliant on the database and changes depending on the size of the database and the image quality.

3. Proposed Framework

3.1. System Overview. The proposed framework for identifying plant diseases utilizes plant leaf images as input. The framework comprises three main processing phases:

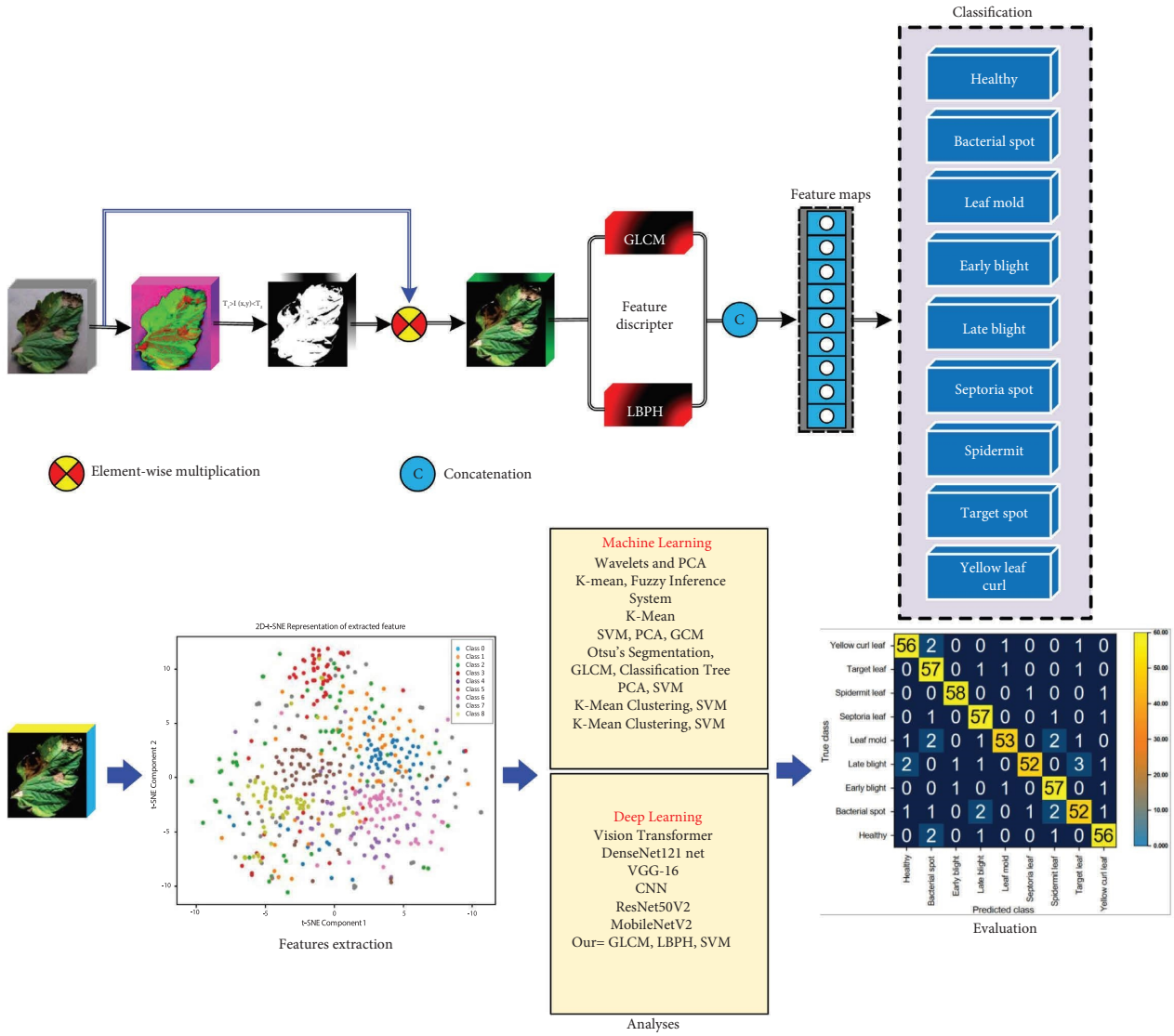

Figure 1: Proposed framework of the plant disease identification system.

classification, feature excision, and foreground segmentation. In the first stage, the input image is improved and segmented using a multithreshold technique, and then GLCM and local binary patterns histogram characteristics are extracted. These characteristics represent specific leaf images during the classification stage. The classification of plant diseases is performed using the support vector machine (SVM) classifier. Figure 1 depicts the complete framework of the proposed network for identifying plant diseases.

3.2. Image Segmentation. Partitioning an image into similar regions based on a connected group of pixels with similar properties with a set of selected measures comprising thresholding and region-growing known similarity-based segmentation [10, 14]. In similarity-based segmentation, thresholding is the most widely used powerful and simplest technique. It is used for background extraction by converting the original image into binary and masking it. In automatic thresholding, the threshold value can be pixel intensities, object size, and probability of occurrence. There are various schemes of automatic thresholding, such as P-tile, iterative, and adaptive thresholding [15]. Here, we are using a technique based on P-tile thresholding that works by setting minimum and maximum ranges of brightness values to select the pixels as foreground within the range and rejects all other pixels as background [16–18]. It is a global thresholding technique based on a specific threshold value, and output depends on a specific threshold value T. It works on single and multiple threshold values denoted by threshold T and threshold values T_0 and T_1, respectively. Segmentation through a single threshold value T is as follows:

$$I_{\text{out}}\left(\mathrm{I}_x, \mathrm{I}_y\right) = \begin{cases} 0, & I_{in}(\mathrm{x}, \mathrm{y}) < \mathrm{T}, \\ 1, & I_{in}(\mathrm{x}, \mathrm{y}) \geq \mathrm{T}, \end{cases} \tag{1}$$

where T stands for the color threshold value, $I_{in}(x, y)$ stands for the original pixel value, and $I_{out}(\mathrm{I}_x, \mathrm{I}_y)$ stands for the resultant pixel value. The input image is encoded into an actual binary image by equation (1). If multiple threshold values exist simultaneously, equation (1) can be transformed into the following equation:

$$I_{\mathrm{out}}\left(\mathrm{I}_x, \mathrm{I}_y\right) = \begin{cases} 0, & I_{\mathrm{in}}(x, y) < \mathrm{T}_1, \\ 1, & T_1 \le I_{\mathrm{in}}(x, y) \le T_2, \\ 0, & I_{\mathrm{in}}(x, y) > \mathrm{T}_2, \end{cases} \tag{2}$$

where T_1 is the lower threshold value, while T_2 represents the upper threshold value because characterizing each pixel in a color image necessitates using multiple variables, one for each red, green, and blue channel. Multispectral thresholding is a viable option for this. However, setting selection criteria for such images is difficult. In this case, a logical extension of the thresholding is to apply a brightness threshold to each image individually, as described in [16, 19]. After converting each image to binary images individually, combine as many of these binary images as possible using the logical AND operator. Color thresholding is considered the most basic and effective segmentation method. The result of image segmentation is depicted in Figure 2

3.3. Feature Extraction

3.3.1. Gray Level Co-Occurrence Matrix (GLCM). The GLCM calculates the special dependencies of gray levels in an image [20, 21]. The number of gray levels in an image directly correlates with the number of columns and rows in the GLCM. The GLCM matrices can be generated in four different orientations ($0^\circ, 45^\circ, 90^\circ$, and 135°). Another matrix is created by averaging the previous matrices. Assuming the co-occurrence matrix is $L(i, j)$ and the matrix size is $(N \times N)$. Each element (i, j) denotes the pixel frequency at a gray level where i is spatially related to the pixel with gray level j. The GLCM signifies the correlation between the reference pixel i and the neighboring pixel j in different orientations [22, 23]. The pixel relationship is calculated horizontally at a degree of (θ°). Initially, all elements (i, j) in the GLCM have a zero value. These element values are then updated based on the occurrence of pixels appearing together. Using the GLCM, various texture features such as energy, contrast, correlation, homogeneity, dissimilarity, ASM, mean, and standard deviation are calculated as follows:

$$\begin{aligned} \mathrm{CONTRAST} &= \sum_{n=0}^{G-1} n^2 \left\{ \sum_{i=1}^{G} \sum_{j=1}^{G} P(i, j) \right\}, |i - j| = n, \\ \mathrm{ASM} &= \sum_{i=1}^{G} \sum_{j=1}^{G} P(i, j)^2, \\ \mathrm{Correlation\,(COR)} &= \frac{\sum_{i=1}^{G} \sum_{j=1}^{G} (i, \overline{x})(\mathrm{j}, \overline{y}) P(i, j)}{\sigma_x, \sigma_y}, \end{aligned} \tag{3}$$

where $\overline{x} = \sum_i^G i \sum_j^G P(i, j)$, $\overline{y} = \sum_j^G j \sum_i^G P(i, j)$, $\sigma_{x^2} = \sum_i^G (i - \overline{x})^2 \sum_j^G P(i, j)$, and $\sigma_{y^2} = \sum_j^G (i - \overline{y})^2 \sum_i^G P(i, j)$.

$$\begin{aligned} \mathrm{Diss} &= \sum_{i=0, j=0}^{G-1} P_{i,j} |i - j|, \\ \mathrm{Energy} &= \sum_{i,j=0}^{G-1} P^2{}_{i,j}, \\ \mathrm{Homg} &= \sum_{i,j=0}^{G-1} \frac{P_{i,j}}{1 + (i - j)^2}, \\ \mathrm{Std}\sigma_i &= \sqrt{\sigma_i^2}, \sigma_j = \sqrt{\sigma_j^2}, \\ \mathrm{Mean}\mu_i &= \sum_{i,j=1}^{G} i\left(P_{i,j}\right), \mu_j = \sum_{i,j=1}^{G} j\left(P_{i,j}\right). \end{aligned} \tag{4}$$

3.3.2. Local Binary Pattern. The local binary pattern was proposed by the authors in [24] for texture classification. Face recognition, object tracking, and image classification all use LBP descriptor features [24, 25]. LBP is mathematically represented as follows:

$$\begin{aligned} \mathrm{LBP}_{p,r} &= \sum_{l=0}^{p-1} 2^l \times S_1\left(I_l - I_c\right) \\ S_1(x) &= \begin{cases} 1, & x \ge 0, \\ 0, & \text{else}, \end{cases} \end{aligned} \tag{5}$$

where I_l and I_c are neighborhood and center pixel values, respectively, for p neighborhood and r radius.

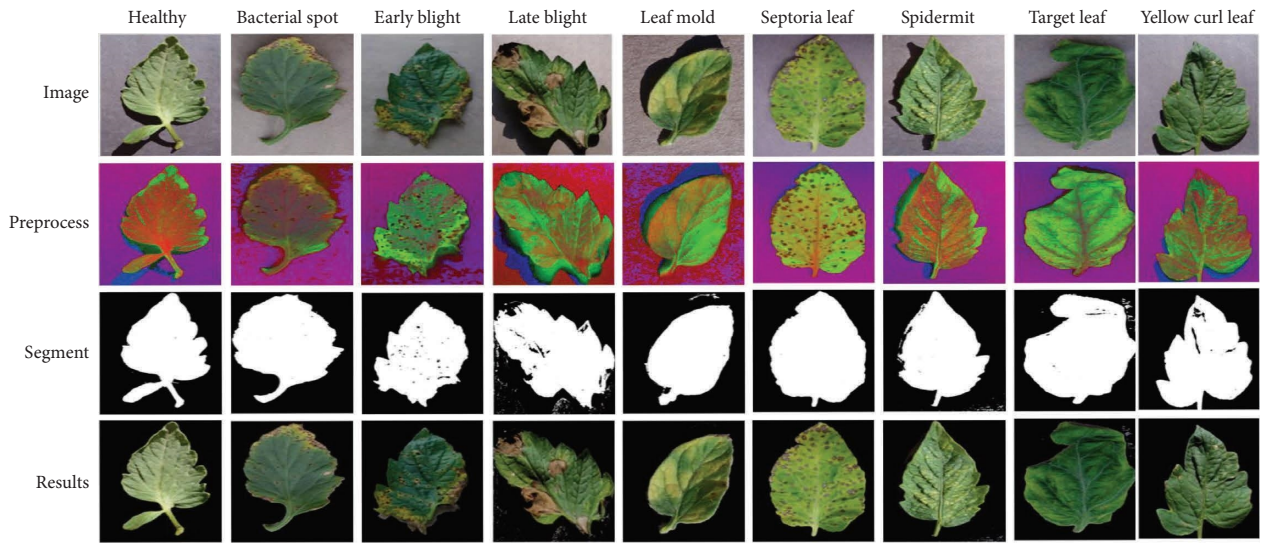

FIGURE 2: During our experiments, we used different samples of images. The sequence of images in each column represents the nine classes of the tomato dataset. The first row shows the original images, the second row displays the preprocess samples, the third row provides information about the segmented images, and the last row demonstrates the final results.

3.3.3. Local Binary Pattern Histogram. The local binary pattern histogram (LBPH), proposed in 2006 [26], is the extension of the local binary pattern (LBP) and is a well-known, widely used algorithm for image recognition due to its computational power and simplicity. As the functionality of LBPH, the image is split into similar-sized regions, and the LBP operator is applied to the region. The LBP operator compares eight closest neighbor pixels from a single point. The comparison replaces lower neighboring numbers from the center to 0 and higher to 1, resulting in an 8-bit binary number. That binary number translated into a single decimal value ranging from 0 to 255, known as a pixel LBP value, is shown in Figure 3.

The histogram is then produced by counting the instances of each LBP value in a certain area. This procedure results in a histogram with 256 bins for each region. The equation below outlines the representation of this method:

$$H_i = \sum_{x,y} I\{LBP(J(x,y)) = i\}, \quad i = 0, \ldots, 255, \tag{6}$$

where the conditional operator I returns 1 if the statement is true or 0 if it is false, and H_i is the bin of value i, $J(x, y)$ and (x, y) is the pixel of the image. After calculating the histograms for each region, the histograms for each part are combined to form a single histogram. The feature vector of the image is considered as the final histogram, which has $(256 \times m \times m)$ bins.

3.3.4. Classification: Support Vector Machine (SVM). The support vector machine (SVM) is a highly efficient, widely used machine learning (ML) algorithm for classification and regression problems. SVM resolves the classification problem by finding the best separating hyperplane among different classes. SVM seeks to maximize the boundary around a hyperplane that splits a negative class from a positive one. Given a training dataset with n samples (a_1, b_1), $(a_2, b_2), \ldots..$, (a_n, b_n), where a_i is a feature vector with m-dimensional feature space. And the labels $b_i \in 1, 1$ belong to any of the two classes C_1 and C_2, which are separable. Equations (7) and (8) demonstrate how the SVM process employs an ideal hyperplane with the maximum boundary to distinguish between two classes and solve the optimization issue.

$$\text{maximize} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \propto_j b_i b_j \cdot K\left(a_i a_j\right), \tag{7}$$

$$\text{Subject} - \text{to}: \sum_{i=1}^{n} \propto_i b_i, \quad 0 \leq \propto_i \leq C, \tag{8}$$

where α_i is the assigned weight to the training sample a_i. If $\alpha_i > 0$, a_i is called a support vector. A regulation parameter C is used to trade off the training accuracy and model complexity to achieve superior generalization capability. Using kernel function K, the similarity between two samples is measured. Although many kernel functions are available, the most popular ones are the liner, Gaussian, radial basis function (RBF), and polynomials of a certain degree. Regardless of the challenge, these kernels are typically applied to continuous and discrete data.

3.3.5. Performance Evaluation Measures. Different evaluation measures are used in our analysis, which include precision (P_r), recall (R_c), accuracy (Acc), F-measure (F_m), and confusion matrix (Cmat). These measures are defined using specific terms such as false positive (FP), true positive (TP), true negative (TN), and false negative (FN). P_r decreases when false positives (FPs) increase. The R_c measures the correct prediction of positive cases by calculating the proportion of true positives (TPs) to the total sample

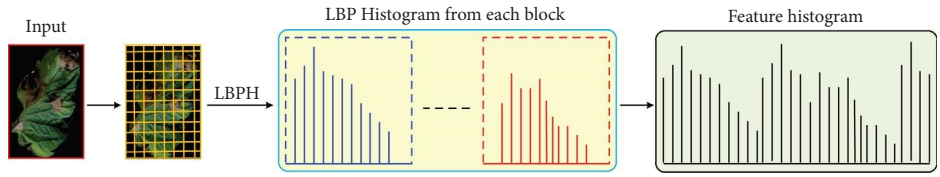

Figure 3: Overview of LBPH.

Table 1: Overview of the used dataset.

Sr. no.	Name	No. of images
1	Bacterial spot	300
2	Early blight	300
3	Healthy	300
4	Late blight	300
5	Leaf mold	300
6	Septoria leaf spot	300
7	Spidermit	300
8	Target spot	300
9	Yellow curl leaf	300
Total		2700

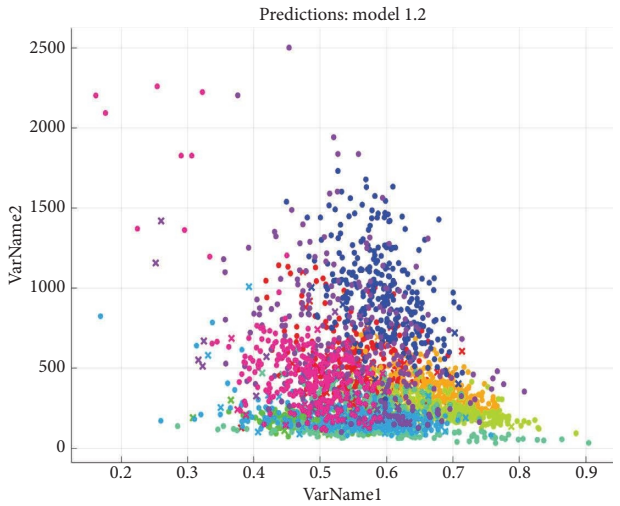

Figure 4: The scatter plot displays features from nine classes. It shows the correlations between the different classes and their respective features.

Table 2: Classification results.

Validation scheme	SVM kernel	Accuracy (%)
10-Fold cross-validation	Linear	87.5
	Quadratic	**92.3**
	Gaussian	88.3
	Cubic	91.7
70 – 20% validation	Linear	86.9
	Quadratic	**91.6**
	Gaussian	87.5
	Cubic	90.1

The bold values represent the highest performance metrics.

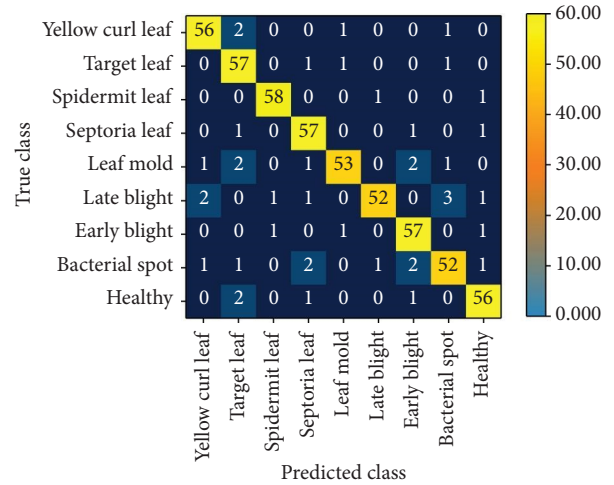

Figure 5: Confusion matrix of the proposed approach.

(TP + FN). The range of P_r and R_c is between 0 and 1. The F_m evaluates the model's performance by calculating the weighted harmonic mean of precision P_r and R_c. This classification is based on multiclassification [27], and the dataset's total number of sample sizes is indicated by M (class = 9). Mathematically, all these evaluation measures are defined using the following equations:

$$P_r = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{9}$$

$$R_c = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{10}$$

$$F_m = 2 \times \frac{(\text{precision}) \times (\text{recall})}{(\text{precision}) + (\text{recall})}, \tag{11}$$

$$A_{cc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{12}$$

4. Analysis of Results

The experimental results in this study were obtained using a computer system equipped with an Intel Core i7 @ 2.2 GHz CPU and 16 GB of memory. In addition, we utilized Python OpenCV software on the Windows XP 10 operating system. The dataset used for this research was collected from online resources (https://plantvillage.org/). This dataset comprises

Table 3: Comparison of various machine learning and deep learning architectures using the tomato leaf dataset, and the symbol (〃) is used to indicate that the dataset has the same name.

Title	Methods	No. of images	Accuracy (%)	No. of classes	
Identification of leaf diseases in tomato plant based on wavelets and PCA [28]	Wavelets and PCA	50	90	4	Machine learning
Identification of black mold disease in tomato using the black system [29]	K-means and fuzzy inference system	100	87	3	
Tomato disease segmentation using K-means clustering [28]	K-means	200		5	
Automatic detection of diseased tomato plants using thermal and stereo visible light images [30]	SVM, PCA, and GCM	400	90	2	
Tomato plant disease classification in digital image using classification tree [31]	Otsu's segmentation, GLCM, and classification tree [31]	383	97.3	5	
Multiclass SVM-based classification approach for tomato ripeness [32]	PCA and SVM	235	92.72	5	
Tomato leaf diseases' detection approach based on support vector machines [33]	K-means clustering and SVM	200	99.5	2	
Analysis of late blight disease in tomato leaf using image processing techniques	K-means clustering and SVM	100	84	2	
Tomato leaf disease classification (https://plantvillage.org/)	Vision transformer [34]	2700	89	9	Deep learning
〃	DenseNet121_net [35]	2700	79	9	
〃	VGG-16 [36]	2700	73	9	
〃	CNN [37]	2700	78	9	
〃	ResNet50V2 [38]	2700	76	9	
〃	MobileNetV2 [39]	2700	91	9	
Ours	GLCM, LBPH, and SVM	**2700**	**92.3**	**9**	

The bold values represent the results from our proposed method, which includes GLCM, LBPH, and SVM, highlighting the performance of our approach.

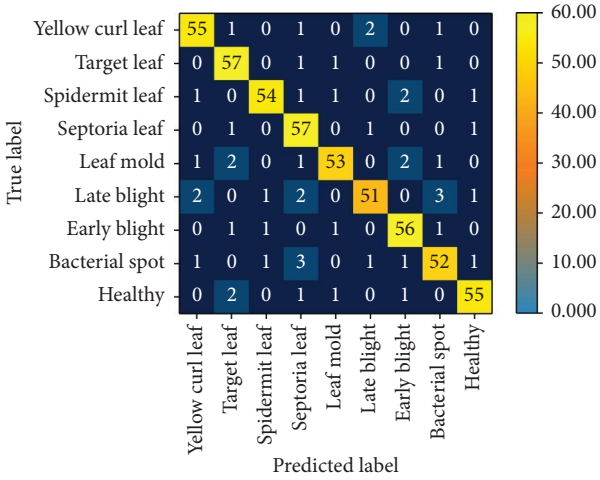

Figure 6: Confusion matrix of MobileNetV2.

2,700 images of tomato plant leaves, categorized into 9 classes, and the comprehensive information is given in Table 1. The classes are healthy, bacterial spots images, early blight images, Spidermite, leaf mold, late blight, Septoria leaf spot, target spot, and yellow leaf color virus. The proposed framework used a variety of SVM kernels, including linear, quadratic, Gaussian, and cubic kernels for feature classification. We used 10-fold cross-validation and 80 – 20% training testing techniques to validate the classification findings. Energy, contrast, dissimilarity, correlation, mean, homogeneity, Minpix, standard deviation, Maxpix, coefficient of variance, and scale-invariant feature transform (SIFT) comprise the combined feature vector of GLCM texture features. Figure 4 shows scatter plots of the dataset, while Table 2 presents the outcomes obtained by utilizing various SVM kernels. The confusion matrix in Figure 5 provides a comprehensive overview of the accuracy of individual classes in the case of the SVM classifier with a cubic kernel. This matrix allows us to assess the classifier's performance by displaying the number of correctly classified instances for each class and any misclassifications. In our specific tomato plant disease classification case, we employed various machine learning and deep learning models. Out of all the models, the SVM classifier achieved an impressive accuracy of 92.3% as presented in Table 3. This means that the model correctly identified the disease status of tomato plants, showcasing its effectiveness in distinguishing between different disease classes. This high accuracy rate indicates the potential of the SVM classifier with a cubic kernel as a reliable tool for automated tomato plant disease identification and management.

4.1. Confusion Metric. To evaluate the performance of the proposed method, a confusion matrix was used to analyze the results of the testing phase, as shown in Figure 5. The x-axis represents the predicted label, while the y-axis represents the actual label. In this study, the author compared the true label of the tomato leaf dataset with the predicted label. The yellow curl leaf and healthy classes achieved an accuracy of 93%. The target leaf, Septoria leaf, and early blight classes also performed well, with an accuracy of 95%. Notably, the Spidermit leaf class stood out among all classes with an outstanding accuracy of 96% during the testing phase. However, the late blight and bacterial spot classes had a slightly lower accuracy of 86%, indicating that the proposed method had some difficulty in correctly classifying these classes, while leaf mold obtained an accuracy of 88%. These results demonstrate the overall effectiveness of the proposed method in accurately classifying the different classes of the tomato leaf dataset during the testing phase.

Similarly, we also trained and tested the same dataset on several deep learning models, including ViT, ResNet, VGG, and DenseNet121. However, MobileNetV2 emerged as the top performer, surpassing the other models in terms of accuracy and prediction performance. The confusion matrix of MobileNetV2, displayed in Figure 6, provides a visual representation of the model's predictions. It showcases the distribution of true-positive, true-negative, false-positive, and false-negative values, allowing us to analyze the model's effectiveness in classifying the dataset. During the testing phase, both the yellow curl leaf and healthy classes achieved an accuracy of 91%. The target leaf class had the highest accuracy at 95%, indicating that the model is exceptionally proficient at identifying this class. The Spidermit leaf class achieved an accuracy of 90%, while the leaf mold class had an accuracy of 88%. The late blight class achieved 85% accuracy, which is still quite good considering the complexity of this class. The early blight and bacterial spot classes achieved accuracies of 93% and 86%, respectively, showcasing the model's ability to accurately classify these classes as well. Overall, the confusion matrix and the individual class accuracies demonstrate that the proposed approach performs exceptionally well in classifying the tomato leaf dataset.

5. Conclusions

The detection and classification of tomato crop diseases are crucial for farmers to effectively manage and control the health of their plants. This task can be made more accurate and time-efficient by employing efficient techniques, such as image processing. Image processing involves analyzing digital images of tomato plants to identify disease symptoms or patterns, such as leaf discoloration, spots, or deformities. This technology enables farmers to detect and classify diseases quickly, allowing for timely intervention and targeted treatment. By leveraging image processing techniques, farmers can enhance the effectiveness of their disease management strategies, minimize crop losses, and ensure the overall health and productivity of their tomato crops. In this study, we utilized images of plant leaves exhibiting visual symptoms of specific diseases. To classify the various defects in the image dataset, we employed an SVM-based approach with GLCM and SIFT features. By incorporating both GLCM and SIFT features, we can leverage their complementary strengths to enhance the accuracy and robustness of the classification model. Our proposed framework extracts valuable features from the image and effectively classifies different disease types. In our experiments, we achieved an

impressive accuracy of 92.3%. Currently, our system focuses on texture features, but in future work, we plan to expand it by incorporating shape- and color-based features in addition to texture features. This will further enhance the performance of our classification model.

Data Availability

In all of the experiments, the NVIDIA GPU GTX-1070 was employed. This model was created on a GPU with 64 GB of memory and an Intel i9 9900k processor. The development process was completed using PyTorch. The dataset used for this research was collected from online open resources (https://plantvillage.org/). Code availability: The custom codes used to produce the results presented in this paper are available from the corresponding authors upon reasonable request.

Disclosure

The funders did not influence the study design, data collection, analysis and interpretation, manuscript writing, and publication decisions.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Authors' Contributions

Rashid Khan and Nasir Ud Din conceptualized and designed the study, was involved in data acquisition, analyzed and interpreted the data, and prepared the draft. Asim Zaman wrote the manuscript and reviewed and edited the manuscript. Bingding Huang investigated the data, reviewed and edited the manuscript, and supervised the study. Rashid Khan and Nasir Ud Din have contributed equally to this work.

Acknowledgments

This work was supported by the Project of the Educational Commission of Guangdong Province of China under Grant no. 2022ZDJS113.

References

[1] E. Loizou, C. Karelakis, K. Galanopoulos, and K. Mattas, "The role of agriculture as a development tool for a regional economy," *Agricultural Systems*, vol. 173, pp. 482–490, 2019.

[2] A. Romanenko, "World - tomatoes - market analysis, forecast, size, trends and insights," 2021, https://www.indexbox.io/store/world-tomato-market-reportanalysis-and-forecast-to-2020/.

[3] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, "Plant leaf disease detection using computer vision and machine learning algorithms," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 305–310, 2022.

[4] M. Á. Rodríguez-García, F. García-Sánchez, and R. Valencia-García, "Knowledge-based system for crop pests and diseases recognition," *Electronics*, vol. 10, no. 8, p. 905, 2021.

[5] Q. Wang, D. Wu, W. Liu et al., "PlantStereo: a high quality stereo matching dataset for plant reconstruction," *Agriculture*, vol. 13, no. 2, p. 330, 2023.

[6] T. Saranya, C. Deisy, S. Sridevi, and K. S. M. Anbananthen, "A comparative study of deep learning and Internet of Things for precision agriculture," *Engineering Applications of Artificial Intelligence*, vol. 122, Article ID 106034, 2023.

[7] N. U. Din, L. Zhang, and Y. Yang, "Automated battery making fault classification using over-sampled image data cnn features," *Sensors*, vol. 23, no. 4, p. 1927, 2023.

[8] V. Singh, A. Misra, and Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Information Processing in Agriculture*, vol. 4, no. 1, pp. 41–49, 2017.

[9] H. Orchi, M. Sadik, and M. Khaldoun, "On using artificial intelligence and the internet of things for crop disease detection: a contemporary survey," *Agriculture*, vol. 12, no. 1, p. 9, 2021.

[10] V. Moysiadis, P. Sarigiannidis, V. Vitsas, and A. Khelifi, "Smart farming in europe," *Computer science review*, vol. 39, Article ID 100345, 2021.

[11] A. Haridasan, J. Thomas, and E. D. Raj, "Deep learning system for paddy plant disease detection and classification," *Environmental Monitoring and Assessment*, vol. 195, no. 1, p. 120, 2023.

[12] S. Sural, Q. Gang, and S. Pramanik, "Segmentation and histogram generation using the hsv color space for image retrieval," in *Proceedings of the International Conference on Image Processing*, Rochester, NY, USA, September 2002.

[13] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and Trends in Computer Graphics and Vision*, Now Foundations and Trends, Norwell, MA, USA, 2008.

[14] C. Wen and T. Guo, "Segmentation of fingerprint images using minimal graph cuts," in *2009 International Symposium on Computer Network and Multimedia Technology*, Wuhan, China, January 2009.

[15] N. Kulkarni, "Color thresholding method for image segmentation of natural images," *International Journal of Image, Graphics and Signal Processing*, vol. 4, no. 1, pp. 28–34, 2012.

[16] A. Chen, T. Wittman, E. Tartakovsky, and A. Bertozzi, "Image Segmentation through Efficient Boundary Sampling," in *Proceedings of the 2009 Workshop on Sampling Theory and Applications*, Marseille, France, May 2009.

[17] Y. R. Chen, K. Chao, and M. S. Kim, "Machine vision technology for agricultural applications," *Computers and Electronics in Agriculture*, vol. 36, no. 2–3, pp. 173–191, 2002.

[18] C. Wang, Y. Tang, X. Zou, W. Situ, and W. Feng, "A robust fruit image segmentation algorithm against varying illumination for vision system of fruit harvesting robot," *Optik*, vol. 131, pp. 626–631, 2017.

[19] R. Raof, Z. Salleh, S. I. Sahidan et al., "Color thresholding method for image segmentation algorithm of ziehl-neelsen sputum slide images," in *International Conference on Electrical Engineering*, Mexico City, Mexico, November 2008.

[20] S. K. Ps and D. Vs, "Extraction of texture features using glcm and shape features using connected regions," *International Journal of Engineering and Technology*, vol. 8, no. 6, pp. 2926–2930, 2016.

[21] G. Preethi and V. Sornagopal, "Mri image classification using glcm texture features," in *International Conference on Green*

Computing Communication and Electrical Engineering, Coimbatore, India, March 2014.

[22] A. V. Alvarenga, W. Pereira, A. Infantosi, and C. M. Azevedo, "Complexity curve and grey level co-occurrence matrix in the texture evaluation of breast tumor on ultrasound images," *Medical Physics*, vol. 34, no. 2, pp. 379–387, 2007.

[23] X. Zhang, J. Cui, W. Wang, and C. Lin, "A study for texture feature extraction of high-resolution satellite images based on a direction measure and gray level co-occurrence matrix fusion algorithm," *Sensors*, vol. 17, no. 7, p. 1474, 2017.

[24] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[25] T. Ahonen, J. Matas, C. He, and M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 2037–2041, 2009.

[26] M. P Arakeri, M. Arun, and R. K. Padmini, "Analysis of late blight disease in tomato leaf using image processing techniques," *International Journal of Engineering and Manufacturing*, vol. 5, no. 4, pp. 12–22, 2015.

[27] N. U. Din, L. Zhang, Y. Zhou et al., "Laser welding defects detection in lithium-ion battery poles," *Engineering Science and Technology, an International Journal*, vol. 46, Article ID 101495, 2023.

[28] G. M. James and S. Punitha, "Tomato disease segmentation using means clustering," *International Journal of Computer Applications*, vol. 975, p. 8887, 2016.

[29] Y. Mamta, "Identification of black mold disease in tomato using fuzzy inference system," *International Journal for Research in Applied Science and Engineering Technology*, vol. 5, no. 4, pp. 1384–1389, 2017.

[30] S.-e.-A. Raza, G. Prince, J. P. Clarkson, and N. M. Rajpoot, "Automatic detection of diseased tomato plants using thermal and stereo visible light images," *PLoS One*, vol. 10, no. 4, Article ID e0123262, 2015.

[31] H. Sabrol and K. Satish, "Tomato plant disease classification in digital images using classification tree," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1242–1246, IEEE, Melmaruvathur, India, April 2016.

[32] E. Elhariri, N. El-Bendary, M. M. M. Fouad, J. Platoš, A. E. Hassanien, and A. M. Hussein, "Multi-class svm based classification approach for tomato ripeness," in *Innovations in Bio-Inspired Computing and Applications*, pp. 175–186, Springer, Berlin, Germany, 2014.

[33] U. Mokhtar, M. A. Ali, A. E. Hassenian, and H. Hefny, "Tomato leaves diseases detection approach based on support vector machines," in *2015 11th International Computer Engineering Conference (ICENCO)*, pp. 246–250, IEEE, Cairo, Egypt, December 2015.

[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An image is worth 16x16 words: transformers for image recognition at scale," 2020, https://arxiv.org/abs/2010.11929.

[35] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 2017.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[37] M. A. vanallam, "Image-classification-with-cifar-10," 2022, https://github.com/vanallam/Image-classification-with-CIFAR-10.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision–ECCV 2016: 14th European Conference*, Springer, Amsterdam, Netherlands, October 2016.

[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedingsof the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, Salt Lake, UT, USA, June 2018.