



A Novel Approach for Tomato Leaf Disease Classification with Deep Convolutional Neural Networks

Gizem IRMAK^a , Ahmet SAYGILI^{b*}

^aTekirdağ Namık Kemal University, Çorlu Engineering Faculty, Department of Computer Engineering, No:13, 59860 Çorlu, Tekirdağ, TURKEY

^bTekirdağ Namık Kemal University, Çorlu Engineering Faculty, Department of Computer Engineering, No:13, 59860 Çorlu, Tekirdağ, TURKEY

ARTICLE INFO

Research Article

Corresponding Author: Ahmet SAYGILI, E-mail: asaygili@nku.edu.tr

Received: 25 July 2023 / Revised: 25 October 2023 / Accepted: 08 December 2023 / Online: 26 March 2024

Cite this article

Irmağ G, Saygılı A (2024). A Novel Approach for Tomato Leaf Disease Classification with Deep Convolutional Neural Networks. *Journal of Agricultural Sciences (Tarım Bilimleri Dergisi)*, 30(2):367-385. DOI: 10.15832/ankutbd.1332675

ABSTRACT

Computer-aided automation systems for the detection of plant diseases represent a challenging and highly impactful research domain in the field of agriculture. Tomatoes, a major and globally significant agricultural commodity, are cultivated in large quantities. This study introduces a novel approach for the automated detection of diseases on tomato leaves, leveraging both classical machine learning methods and deep neural networks for image classification. Specifically, classical learning methods employed the local binary pattern (LBP) technique for feature extraction, while classification tasks were carried out using extreme learning machines, k-nearest neighborhood (kNN), and support vector machines (SVM). In contrast, a novel convolutional neural network (CNN) framework, complete with unique parameters and layers, was

utilized for deep learning. The results of this study demonstrate that the proposed approach outperforms state-of-the-art studies in terms of accuracy.

The classification process covered various scenarios, including binary classification (healthy vs. unhealthy), 6-class classification, and 10-class classification for distinguishing different types of diseases. The findings indicate that the CNN model consistently outperformed classical learning methods, achieving accuracy rates of 99.5%, 98.50%, and 97.0% for 2-class, 6-class, and 10-class classifications, respectively. Future research may explore the use of computer-aided automated systems to detect diseases in diverse plant species.

Keywords: Automated agriculture, Machine learning in agriculture, Convolutional neural networks in plant pathology, Deep learning in agriculture, Smart farming

1. Introduction

Diseases and pests have a detrimental impact on the production of various agricultural goods, resulting in reduced yields. The timely detection of these diseases and pests is imperative to mitigate the resultant damage. Currently, visual observation by agricultural experts serves as the primary method for detection. The development of computer-aided systems for early disease diagnosis and treatment is proposed to address these challenges.

Computer-aided systems have been widely adopted in agriculture and various other fields. Classical and deep learning methods are frequently employed for the classification of different plant species and disease detection. In recent years, several studies have concentrated on disease detection in various plants, such as tomatoes (Tian et al. 2019; Chen et al. 2020; Karthik et al. 2020; Ouhami et al. 2020; Wspanialy & Moussa 2020; Altuntaş & Kocamaz 2021; Gonzalez-Huitron et al. 2021; Sembiring et al. 2021), rice (Jiang et al. 2020; Sethy et al. 2020; Temniranrat et al. 2021), apple (Park et al. 2018; Kuta et al. 2020; Zhong & Zhao 2020; Rehman et al. 2021), and carrot (Methun et al. 2021) plants. Sustainable agriculture underscores the necessity for efficient, cost-effective, and environmentally friendly techniques. With advancements in computer hardware and software technology, image processing and computer vision have emerged as crucial tools in agriculture, enabling the automated and rapid identification of crop diseases (Xu et al. 2017). The advantages of these technologies include high processing speeds, minimal errors, and improved accuracy. Agricultural research is not limited solely to the automatic detection and classification of crop diseases, it also encompasses other aspects that enhance the overall efficiency of the agricultural sector.

1.1. Related works

In the past 15 years, numerous studies have been undertaken to examine plants and detect diseases by utilizing computer-based systems. Early studies primarily employed classical techniques, including feature extraction, selection, and fundamental image

processing methods. The outcomes generated from these techniques were subsequently incorporated into classical learning methods to facilitate learning processes. More recently, there has been a notable surge in the adoption of deep learning methods, which consolidate all these stages into a unified approach. In general, these research endeavors can be broadly classified into two primary categories: (1) classical learning methods and (2) deep learning methods. In the subsequent section, we will delve into the specifics of studies conducted within these two classifications.

1.2. Classical machine learning-based methods

Liming and Yanchao harnessed machine vision technology to enhance the commercial value of strawberries and classify them within the agricultural sector. They employed the k-means clustering method for strawberry classification and adopted multi-specific decision theory to address multifaceted issues. The study provided impressive results, with strawberries being graded in an average of just 3 seconds. Furthermore, the strawberry size detection error remained below 5%, the color grading accuracy reached 88.8%, and the shape classification accuracy exceeded 90% (Liming & Yanchao 2010). In a related study, Burgos-Artizzu and colleagues presented various image processing-based methods for estimating crop, soil, and weed percentages in cultivation area images. They utilized a genetic algorithm to obtain the optimal combination of method and parameter for different image groups, resulting in an average success rate of up to 96% for winter cereal images and 84% for maize images (Burgos-Artizzu et al. 2010). Adebayo and their team reviewed the application of backscatter imaging to monitor food quality in agriculture, specifically discussing laser light backscatter imaging (LLBI), multispectral laser backscatter imaging (MBI), and hyperspectral laser backscatter imaging (HBI). The study examined the effects of moisture, firmness, acidity, and external defects on agricultural and food quality and emphasized the importance of real-time ranking and evaluation for successful quality assessment (Adebayo et al. 2016). Dutta and colleagues proposed an image-processing method to differentiate untreated and pesticide-treated grapes, addressing the high cost, time-consuming nature, and specialized expertise required for chemical methods of pesticide identification. Their study introduced an effective image processing-based non-destructive method for grape classification, achieving 100% accuracy using a Haar filter and support vector machine (SVM) classifier (Dutta et al. 2016). These studies represent promising tools for quality assessment and highlight the potential for future applications to other agricultural products.

Arakeri and Lakshmana discussed the tomato grading process in India, emphasizing the need for careful handling during grading due to the sensitivity of the fruit. To address this, their study proposed an automatic computer-based system. Various experiments were conducted on tomato images, resulting in the proposed method accurately classifying tomatoes as defective or flawless with 100% accuracy. Additionally, an accuracy of 96.47% was achieved for the classification of ripe and unripe tomatoes (Arakeri 2016). Xu and their team developed a system to rapidly and accurately determine wheat leaf rust (BYR) disease to take timely measures to prevent significant decreases in wheat production. BYR is one of the main fungal diseases that cause crop losses. The system integrated embedded Linux and digital image processing techniques and was successfully implemented on the ARM9 microprocessor. Digital image processing and GUI programs were written and compiled with the help of Qt software for crop disease detection and grading. Results were displayed on an LCD screen (Xu et al. 2017).

Compared to expert diagnoses, the system proposed in the study had accuracy rates of 96.2% and 92.3%, which is close to the decision accuracy of the human eye. In addition, the system was more convenient than human judgment. The proposed system could be used as an agricultural robot to examine field areas and detect, identify, diagnose, and grade all wheat diseases (Liming & Yanchao 2010). In their study, Rahman and colleagues aimed to target disease detection and treatment of tomato plants. The study extracted 13 different statistical features using the GLCM method and classified using the SVM method. The study achieved a 100% success rate for the healthy class, while 95% success was achieved for early blight, 90% for Septoria leaf spot, and 85% for late blight (Rahman et al. 2023). Gerdan and colleagues detected tomato diseases in their study. They used convolutional neural networks (CNN), DenseNet201, InceptionResNetV2, MobileNet, and Visual Geometry Group 16 methods. The most successful method in their study was the CNN method.

Table 1 provides an overview of studies conducted using classical machine learning methods, including information about the plant species, methods employed, and success rates achieved.

Table 1- Studies in the Literature Based on Machine Learning Methods

<i>Author(s)</i>	<i>Plant Type</i>	<i>Methods</i>	<i>Success</i>
Tian et al. 2019	Tomato	k-means algorithm	Accuracy: 90.00%.
Xu et al. 2017	Wheat	Gray level histogram change, edge detection, noise removal.	Recognition rate: 96.20%. Accuracy: 92.30%.
Liming & Yanchao 2010	Strawberry	Segmentation, edge detection, feature extraction, K-means clustering, and classification.	Color grading accuracy: 88.80%. Shape classification accuracy: 90.00%
Burgos-Artizzu et al. 2010	Crop	Segmentation, removal of unwanted regions, genetic algorithm.	Up to 96.00% average success on grain images and 84.00% success on corn images.
Adebayo et al. 2016	Apple	Laser light backscatter imaging (LLBI), multispectral laser backscatter imaging (MBI), and hyperspectral laser backscatter imaging (HBI).	Varying results for LLBI, MBI, and HBI at different wavelengths. For example, in a study conducted on bananas with LLBI in 2013, the detection accuracy was 96.10% at 532 nm wavelength.
Dutta et al. 2016	Grape	Haar filter, segmentation, feature extraction, SVM.	100.00% accuracy was achieved with the SVM classifier.
Arakeri 2016	Tomato	Segmentation, feature extraction, and selection, artificial neural networks.	Accuracy for imperfect and perfect classification: 100.00%. Accuracy for mature and immature classification: 96.47%.
Rahman et al. 2023	Tomato	GLCM, SVM	Accuracy for: Healthy: 100.00% Early blight: 95.00% Septoria leaf spot: 90.00% Late blight: 85.00%
Gerdan et al. 2023	Tomato	CNN, DenseNet201, InceptionResNetV2, MobileNet, Visual Geometry Group 16	CNN Acc.: 99.82%, DenseNet201 Acc.: 92.12%, InceptionResNetV2 Acc.: 92.75%, MobileNet Acc.: 91.50% Visual Geometry Group 16 Acc.: 84.12%

1.3. Deep learning-based methods

Brahimi and colleagues conducted a classification process on tomato leaves with nine different diseases, utilizing a dataset containing 14,828 images. They employed the CNN method for classification and achieved an impressive accuracy rate of 99.18%. The authors suggested that this method can serve as a practical tool for farmers to protect tomatoes against diseases (Brahimi et al. 2017). In the study by Durmuş et al., the AlexNet and SqueezeNet CNN models were tested using a GPU. The AlexNet model achieved 95.65% accuracy, while the SqueezeNet model achieved 94.3% accuracy, slightly outperforming AlexNet. The SqueezeNet model, which was nearly 80 times smaller than AlexNet, is considered a suitable choice for mobile deep learning classification due to its lightweight nature and low computing requirements. Furthermore, using a smaller network reduces data costs and enhances update rates when mobile applications are updated via mobile communication (Durmuş et al. 2017). Prajwala and their team identified 10 different diseases in tomato crops and utilized LeNet, a convolutional neural network model, in their study. They emphasized that this method, offering an accuracy rate of 94-95%, can assist farmers in accurately identifying leaf diseases with minimal computational effort (Tm et al. 2018). In the study by Karthik et al., deep learning architectures were applied to detect infections in tomato leaves. They first implemented residual learning and then applied the attention mechanism. The study used a dataset featuring three diseases, namely early blight, late blight, and leaf blight on tomato leaves. The proposed approach achieved an overall accuracy of 98%, thanks to the features learned by CNN (Karthik et al. 2020). Ouhami and their team aimed to determine the most suitable machine-learning model for detecting tomato crop diseases in RGB images. They utilized transfer learning models, including DensNet161, DensNet121, and VGG16. The study targeted the automatic detection of six different plant diseases. The results showed that DensNet161 and DensNet121 achieved an accuracy of 94.93%, while VGG16 achieved an accuracy of 90.58% (Ouhami et al. 2020). Sembiring and their team developed a deep-learning model based on the CNN architecture baseline for detecting tomato leaf diseases. The study aimed to classify ten classes of tomato leaves, including one healthy class and nine leaf diseases. They compared their model with VGG Net, ShuffleNet, and SqueezeNet architectures, demonstrating that the proposed architecture outperformed existing architectures with an accuracy of 97.15% (Sembiring et al. 2021).

Bhandari and their colleagues classified tomato leaves into nine different infectious diseases (bacterial spot, early blight, etc.) in their study. They used the EfficientNetB5 transfer learning method in their research and reported an average training accuracy

of 99.84% and an average testing accuracy of 99.07% (Bhandari et al. 2023). Tian and their team proposed a model to identify diseases using tomato leaf images. In this study, three different deep learning network architectures (VGG16, Inception_v3, and Resnet50) were used, and an Android application was also developed. The application can identify tomato diseases with a test accuracy of 99% (Tian et al. 2022).

It is worth noting that there are more studies employing deep learning approaches in the literature than those mentioned above. Table 2 presents an overview of the studies reviewed.

Table 2- Studies in the Literature Based on Deep Learning Methods

<i>Study</i>	<i>Plant Type</i>	<i>Methods</i>	<i>Success</i>
Chen et al. 2020	Tomato	ABCK-BWTR and B-ARNet Model	Accuracy: 89.00%.
Wspanialy & Moussa 2020	Tomato	CNN (U-Net)	PlantVillage dataset accuracy: 97.00%. Left-out-disease dataset accuracy: 98.70%.
Gonzalez-Huitron et al. 2021	Tomato	CNN implemented in Raspberry Pi 4	MobileNetV2 accuracy: 75.00%. NasNetMobile accuracy: 84.00%. Xception accuracy: 100.00%.
Karthik et al. 2020	Tomato	Residual CNN	Overall accuracy: 98.00%.
Altuntaş & Kocamaz 2021	Tomato	Concatenated Deep Features	Overall accuracy: 96.99%.
Ouhami et al. 2020	Tomato	DensNet161, DensNet121, VGG16	DensNet161 accuracy: 95.65%. DensNet121 accuracy: 94.93%. VGG16 accuracy: 90.58%.
Sembiring et al. 2021	Tomato	CNN	Accuracy: 97.15%.
Jiang et al. 2020	Rice	CNN + SVM	CNN + SVM accuracy: 96.80%. AlexNet accuracy: 93.79%. VGG accuracy: 91.65%.
Sethy et al. 2020	Rice	CNN (ResNet50) + SVM	Accuracy: 98.38%
Temniranrat et al. 2021	Rice	CNN (YOLOv3)	Average true positive rate: 95.60%.
Park et al. 2018	Apple	CNN	Sensor type: Hyper-spectral Classifier: Local norm. ReLU accuracy: 86.11%.
Kuta et al. 2020	Apple	Manual Harvesting	Optimal height: 1000 mm. The highest surface pressure picks heights: 500 mm and 2000 mm.
Zhong & Zhao 2020	Apple	CNN (DenseNet-121)	Accuracy: 93.71%.
Rehman et al. 2021	Apple	Resnet-50, MASK RCNN	Accuracy: 96.60%.
Methun et al. 2021	Carrot	Inception v3	Accuracy: 97.40%.
Durmuş et al. 2017	Tomato	CNN (AlexNet and SqueezeNet)	AlexNet accuracy: 95.65%. SqueezeNet accuracy: 94.30%.
Tm et al. 2018	Tomato	CNN (LeNet)	Accuracy: 94.00% -95.00%.
Brahimi et al. 2017	Tomato	CNN (Alexnet and GoogleNet)	AlexNet accuracy: 98.66%. GoogleNet accuracy: 99.18%.
Bhandari et al. 2023	Tomato	EfficientNetB5	Accuracy: 99.07%
Tian et al. 2022	Tomato	VGG16, Inception_v3, and Resnet50	Accuracy: 99.00%

This study is primarily concerned with the detection and classification of tomato leaf diseases, employing both classical and CNN deep learning techniques. The study provides several significant contributions, including the proposal of an effective and robust method for detecting tomato leaf diseases, the development of an original CNN model, the utilization of classical learning methods in a novel approach, the demonstration of the proposed method's versatility across various class numbers (2, 6, and 10), and the ability to adapt the proposed method to diverse plant datasets. The study also reports more favorable results compared to the previous literature. The suggested method has the potential to identify tomato diseases at an early stage and reduce harm by enabling timely disease treatment.

The contributions of this study to the detection and classification of tomato leaf diseases, utilizing classical learning methods and CNN deep learning methods, are outlined as follows:

1. An effective, successful, and robust method to detect tomato leaf diseases was proposed by analyzing images of infected tomato leaves.

2. An original CNN model was created and implemented within the scope of the study.
3. Classical learning methods were used with a unique approach.
4. A robust approach for tomato leaf disease is presented for different classes (2, 6, and 10 classes), and more successful results were obtained than other studies in the literature. To the best of our knowledge, this study is the first attempt to detect disease in tomato leaves using different class numbers (2, 6, and 10 classes).
5. The proposed method is independent of the dataset, making it applicable to different plants and datasets.
6. This study will help to detect tomato diseases early and minimize the damage to farmers by ensuring measures can be taken to treat the diseases on time.

Overall, the study provides valuable insights into the detection and classification of tomato leaf diseases and underscores the effectiveness of both classical learning methods and deep learning methods in this domain.











2. Material and Methods

In this section of the study, comprehensive explanations are given regarding the tomato dataset used, as well as the technical intricacies of classical and deep learning methods. The stages of classical methods are elaborated sequentially, providing a step-by-step understanding of their application. Additionally, each layer of the CNN model employed in our research is individually described, including associated parameters and configurations.

2.1. Tomato dataset

The tomato dataset utilized in this study is an open-access repository sourced from Kaggle, featuring a total of 18 345 images categorized into 10 distinct classes. These categories encompass nine disease categories and one category representing healthy tomato leaves [26]. The tomato dataset encompasses all major leaf diseases that can have a substantial impact on tomato production. Each of these leaf diseases has distinct underlying causes, necessitating different strategies, such as fertilization, spraying, and other interventions, to combat them effectively. The tomato dataset employed in this study (Lamrahi 2021) classifies these diseases into nine distinct types, and includes one category for healthy tomato leaves, as detailed in Table 3.

Table 3- Disease types and properties of Tomato Dataset

<i>Disease Type</i>	<i>Image</i>	<i>Properties of the Diseases</i>
Mosaic Virus		The presence of light green, yellow, and dark green mosaic stains on the leaves is indicative of a disease caused by the tobacco mosaic virus and its various strains or breeds.
Two-Spotted Spider Mite		Red spiders have the potential to reproduce in areas where tomatoes are cultivated, whether in a greenhouse or an open field. Without prompt and effective measures, the infestation of red spiders can lead to significant losses in tomato crops.
Leaf Mold		The infestation of red spiders can sometimes spread to cover entire leaves, resulting in a substantial reduction in crop yield. Conditions that contribute to the prevalence of this disease include sudden temperature changes, excessive humidity, and the presence of shadowless greenhouses.
Septoria Leaf Spot		While this disease primarily affects leaves, it can also manifest on the stems, and flower stalks of plants. It typically begins as small yellowish areas on the leaves and subsequently changes in color, taking on a gray or brown appearance.
Bacterial Spot		Small brown to black spots may develop on the leaves, stems, flowers, and fruit stems. As the disease advances, these small spots have the potential to merge and form larger spots on the leaves. Furthermore, you may observe small, dark brown superficial blisters or lesions on the fruits.
Early Blight		The initial symptoms of the disease are typically observed on older leaves. These symptoms manifest as light green or yellowish-brown spots with a yellow halo. Small spots, each with a diameter of around 0.5 mm, gradually merge to cover the surface of the leaf. As the disease advances, the affected leaves may wither and die, leading to a deterioration in the quality of the fruit due to sun damage.
Leaf Curl Virus		In general, symptoms such as leaf shrinkage, blistering, inward curling, and deformity can be observed. Additionally, the leaves may exhibit yellowing starting from the edges, along with varying degrees of discoloration between the veins, ultimately resulting in an overall yellowed appearance.
Target Spot		The initial symptoms on the leaves present as small, misshapen, and greasy spots. In the later stages, these spots can merge and cause the entire leaf to dry out.
Late Blight		This disease results in the development of pale green to brown spots on the leaves, and occasionally, purplish spots may also appear. The edges of the leaf spots might exhibit a pale green color or show signs of waterlogging.
Healthy		Healthy leaves typically exhibit a proper, undistorted shape and maintain a vibrant green color.

2.2. Preprocessing for classical learning methods

To improve classification accuracy, our study initially applied preprocessing steps to the images used. As illustrated in Figure 1, the first step involved converting the images into grayscale. The images in the tomato dataset are originally in color. The subsequent preprocessing step was noise removal since it is highly unlikely that real-world images are entirely noise-free. Therefore, noise removal is commonly carried out as a preliminary step in many studies. In our research, we utilized the Wiener method for noise removal following grayscale transformation (Lim & Oppenheim 1979). The Wiener method is proficient in reducing image blur and is defined by the following formula:

$$W(u, v) = \frac{H(u, v)}{|H(u, v)|^2 + S_{nx}(u, v)} \quad (1)$$

$S_{nx}(u, v)$ is the signal-to-noise ratio and $H(u, v)$ is the sinc function of the target pixel.

During the preprocessing stage, image sharpening was performed on the images, followed by the application of contrast enhancement. In Figure 1, the preprocessing steps are applied to an image from the bacterial spot class in the dataset. Upon closer inspection, it becomes evident that the preprocessing steps significantly enhance the quality of the original image.

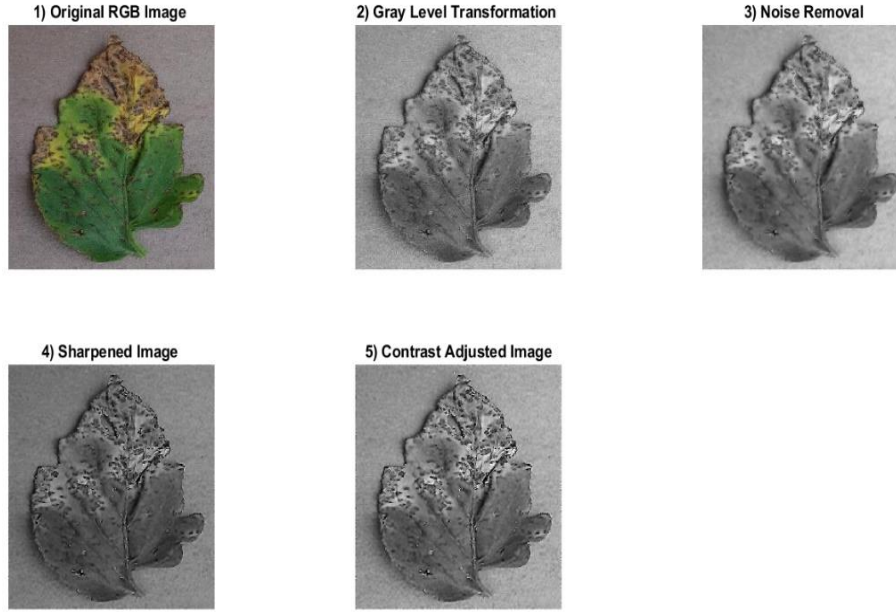


Figure 1- Demonstration of the preprocessing steps in a sample leaf image

2.3. Feature extraction for classical learning methods

In this study, the local binary pattern (LBP) method was employed as a nonparametric feature extraction technique (Ojala et al. 2000). The core principle of this method involves assessing the relationships between pixels by analyzing their neighborhood associations. The computation of LBP is carried out using Equation 2:

$$LBP_{P,R}(x_c) = \sum_{p=0}^{P-1} u(x_p - x_c) 2^p, \quad u(y) = \begin{cases} 0, & \text{if } y < 0 \\ 1, & \text{if } y \geq 0 \end{cases} \quad (2)$$

Where: x_c is the pixel center, x_p is the central pixel neighbors, R is the distance to neighbors, and P is the number of neighbors.

2.4. Classification with classical learning methods

In this stage, the extreme learning machine (ELM), SVM, and k-nearest neighbor (kNN) methods were used. These methods were shown to be effective in various studies in the literature and yield successful results for different class numbers. SVM, developed by Vladimir Vapnik and Alexey Chervonenkis, is a method based on the principle of constructing a hyperplane that separates two classes (Schölkopf & Smola 2002). Here, $(x_i, y_i)_{1 \leq i \leq N}$ indicates training examples, for each example, $x_i \in R^d$, d feature space, y_i class label. The main purpose of SVM is to obtain a hyperplane where samples of the same class will coexist. This hyperplane is expressed with a line equation as in Equation 3.

$$y_i(w \cdot x_i + b) > 0, \quad i=1, \dots, N \quad (3)$$

The kNN algorithm belongs to the category of non-parametric classification methods (Arya et al. 1998). ELM is a model of a single hidden layer feed-forward neural network (SLFN) (Huang et al. 2004, 2006, 2011). The output function of ELM for generalized SLFNs can be expressed as follows:

$$f_L(x) = \sum_{i=1}^L \beta_i h_i(x) = h(x) \beta \quad (4)$$

Where: $\beta = [\beta_1, \dots, \beta_L]^T$ is the output weight vector between the output layer and the hidden layers, $h(x)$ is the output vector of the hidden layer, $h(x) = [G(a_1, b_1, x), \dots, G(a_L, b_L, x)]$ then $G(a, b, x)$ nonlinear piecewise continuous function, and $\{(a_i, b_i)\}_{i=1}^L$ activation functions are randomly generated input values.

Parameter values for all methods used in the study are presented in Table 4.

Table 4- Purpose and Parameters of the Classical Methods

<i>Operation</i>	<i>Operation Type</i>	<i>Parameters</i>
Sharpening	Pre-Processing	Radius:1 Strength of the sharpening effect: 0.7
RGB to Gray Conversion	Pre-Processing	Default
LBP	Feature Extraction	Window Size: 3 x 3 Cell Size: 32
kNN	Classification	k:1 Distance: Minkowski
SVM	Classification	Default
ELM	Classification	Kernel: RBF # of Hidden Neurons: 4096 C: 1e-1

In Figure 2, a structure comprising three fundamental components is observed: preprocessing, feature extraction, and classification, each of which encompasses sub-steps. This figure illustrates the flowchart for the classical learning model employed in our study.

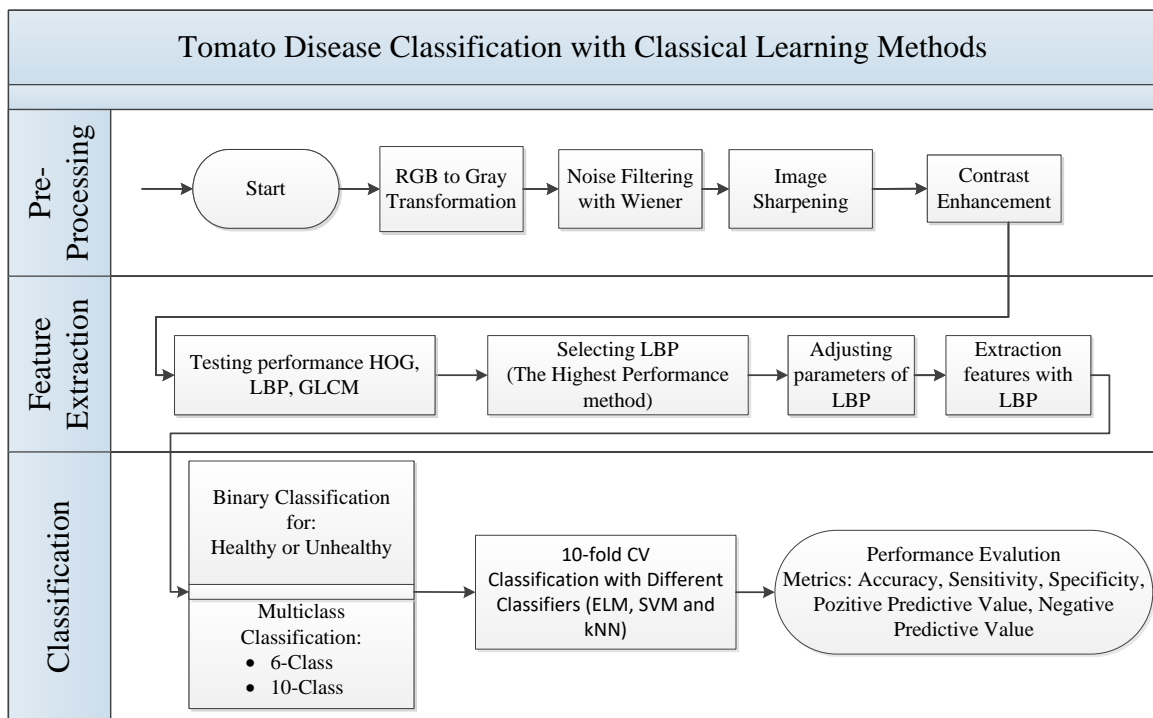


Figure 2- Flowchart for the Process using Classical Learning Methods

As depicted in Figure 2, both preprocessing and feature extraction methods were applied before the utilization of classical learning methods. A novel approach was adopted in terms of the methods used and their application. Various methods with different parameter values were experimentally tested both in feature extraction and preprocessing stages, and the most successful ones were selected. As shown in Figure 2, it is not feasible to perform studies solely on raw data when applying classical methods. Direct classification with raw data can result in very low success rates. Therefore, in classical methods, raw data are pre-processed and subjected to feature extraction, as shown in Figure 2. The success achieved by the system is directly related to the suitability of the feature extraction method.

Feature extraction methods can assess the effectiveness of images based on their structural characteristics. However, it is important to note that the same feature extraction method may not yield similar success across all image datasets. Consequently, the selection of the most successful feature extraction method was determined by comparing the performance of various methods, as evident in the feature selection section of the flowchart above. This choice has a substantial impact on the model's success. Furthermore, during the classification stage, the system was configured for various class numbers using the 10-fold cross-validation (CV) method. This approach contributes to the effectiveness and reliability of the obtained results.

2.5. Classification with deep learning methods

Classical learning methods face limitations when dealing with high-dimensional data, especially when inputs and outputs are extensive. As the complexity of the problem increases, processing such data becomes more challenging in terms of both performance and accuracy. In such scenarios, deep learning provides solutions to intricate real-world problems by constructing more complex network structures of neurons for information transmission.

Deep learning can directly learn features from the data provided. Neural networks can adeptly capture attributes and relationships between data points that may be challenging for other algorithms to discern. By employing layers of neurons that mathematically manipulate data, neural networks can develop complex models. Typically, a neural network model comprises an input layer, an output layer, and one or more hidden layers that facilitate the flow of information between the input and output layers. The term "deep learning" is used to describe models with numerous hidden layers. In Figure 3, each circle represents a neuron, which is a mathematical function with weight, bias, and activation function values. Neurons receive one or more inputs, and information is relayed from the input layer to the hidden layers for processing and, ultimately, to the output layer.

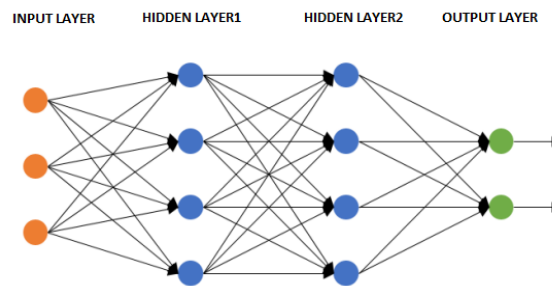


Figure 3- General View of Neural Network Architecture

The activation function plays a critical role in determining whether a neuron in an artificial neural network will be active or not. There are several activation functions to choose from, including "sigmoid", "tanh", "relu" (rectified linear unit), and "SoftMax". The selection of the appropriate activation function depends on the specific problem being addressed. An artificial neuron model is seen in Figure 4. " X_1, X_2, \dots, X_n " are the input values and " W_1, W_2, \dots, W_n " are the corresponding weight values and " b " is a bias value. " f " is the activation function and is the function applied to the value by adding the bias value to the sum of the products of the inputs and the weights.

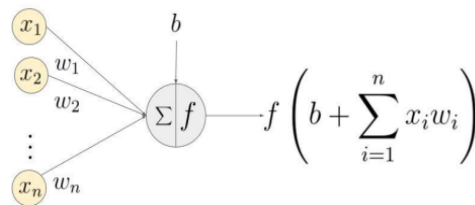


Figure 4- An Artificial Neuron Model

The loss function calculates the disparity between the predicted output and the actual target variable, indicating the level of error. Various error functions, such as "binary cross-entropy" and "negative log-likelihood" can be employed for classification tasks. Throughout the learning process, the goal of the model is to minimize this error to reduce the rate of false predictions. This is achieved by adjusting the weight and bias values, with the ultimate aim of learning the weight and bias values that yield the minimal error rate. Optimization algorithms like "stochastic gradient descent", "Adagrad", "RMSProp" and "Adam" can be used to facilitate this process.

Convolutional neural networks (CNNs) are a subset of deep learning often used for the analysis of visual information, particularly in image and video recognition tasks. They fall under the category of multi-layer neural network models. Compared to classical learning methods, CNNs tend to be more successful and typically do not require additional techniques for feature extraction or preprocessing (Tm et al. 2018). The structure of a CNN, as depicted in Figure 5, includes convolutional, non-linearity, pooling, flattening, and fully connected layers. The convolutional layer is used for feature detection, the non-linearity layer introduces non-linearity to the system, the pooling layer reduces the number of weights and assesses their suitability, the flattening layer prepares the data for the network, and data classification occurs in the fully connected layer.

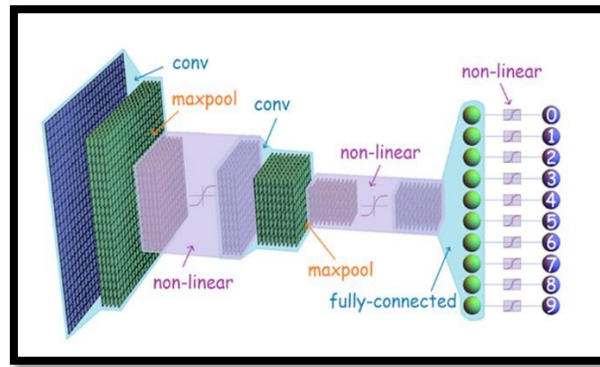


Figure 5- CNN Structure (Anonymous 2021a)

The convolutional layer is the initial layer in CNN algorithms that operates on images. This layer consists of a series of operations between an input 'I' and a set of 'n' convolutional filters ' F^E ', followed by a non-linear activation function. As a result of these operations, the output volume of 0 is expressed in the formula (5).

$$O_m(i, j) = a \left(\sum_{d=1}^D \sum_{u=-2k-1}^{2k+1} \sum_{v=-2k-1}^{2k+1} F_{md}^E(u, v) I_d(i - u, j - v) \right) + b_m \quad (5)$$

Where: ' $2k+1$ ' is the side of a square with an odd convolutional filter, ' a ' refers to the activation function, ' b_m ' refers to the bias for the m^{th} feature map

Each convolutional layer in this architecture is responsible for learning patterns to detect the type of disease in the tomato leaf (Karthik et al. 2020).

The non-linearity layer, also known as the "activation layer," employs one of the activation functions. In the past, nonlinear functions such as "sigmoid" and "tanh" were used; however, the rectifier linear unit (ReLU) function is currently preferred because it provides the best performance in terms of neural network training speed. In this study, the ReLU function was utilized, which is expressed in formula (6) and has a value range of $[0, +\infty]$.

$$f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases} \quad (6)$$

The pooling layer, similar to the convolutional layer, aims to reduce dimensionality. This reduction not only saves processing power but also filters out irrelevant features, emphasizing the more important ones. CNN models often use two primary pooling techniques: maximum (Max) pooling and average pooling. In this study, the maximum pooling technique was used. The image size after the pooling process is calculated as follows:

$$\text{Size of the generated image} = W_2 \times H_2 \times D_2 \quad (7)$$

$$W_2 = (W_1 - F) / N + 1 \quad (8)$$

$$H_2 = (H_1 - F) / N + 1 \quad (9)$$

$$D_2 = D_1 \quad (10)$$

Where: W_1 = Width value of input image size, H_1 = Height value of input image size, D_1 = Depth value of input image size, F = Filter size, N = Number of steps, In the pooling process, $F = 2$ and $N = 2$ were chosen.

The flattening layer is responsible for preprocessing data for the subsequent "fully connected layer." Neural networks typically accept input data in the form of one-dimensional arrays. The data used in this layer is in the form of a one-dimensional matrix obtained from the convolution and pooling layers.

The fully connected layer transforms the matrix image, which has passed through the convolutional and pooling layers multiple times, into a flattened vector.

In the classification layer, the Softmax activation function was chosen for the output layer of our CNN model. This choice was made because more than two classifications (10 classes) were required for tomato leaf diseases.

2.6 Our application of CNN

In our study, the proposed approach consists of three significant stages: data acquisition, data pre-processing, and classification, as depicted in Figure 6.

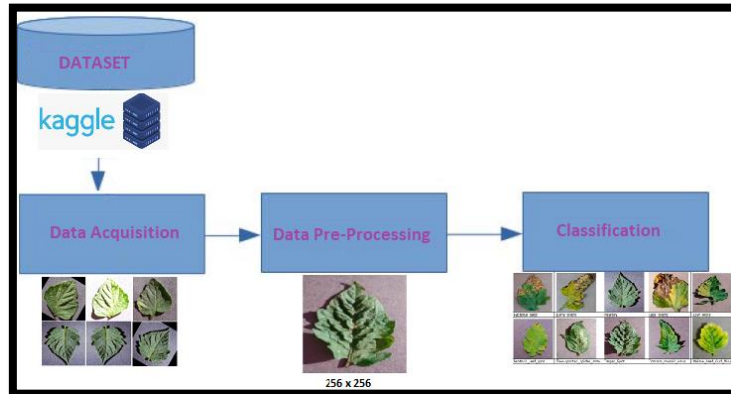


Figure 6- Flowchart for the Process with the Deep Learning Methods

In our study, a tomato leaf dataset sourced from the publicly accessible Kaggle platform was utilized, containing 18 345 images categorized into 10 distinct classes, including 9 categories representing diseased leaves and 1 category for healthy leaves. These categories cover all leaf diseases that can impact tomato production. To improve modeling efficiency and reduce processing time, we resized the images to 256 x 256 resolution. The classification process was carried out using a deep-learning CNN model. The study was conducted in the Python programming language within the Google Colaboratory (Colab) Notebook environment and employed Python libraries such as OpenCV, Keras, Numpy, Os, Sklearn, and Matplotlib.

The dataset included a total of 18 345 images, with approximately 2 000 images per class. For binary classification, 500 samples were randomly selected for both the healthy and unhealthy classes. For the 6-class classification, we chose around 167 samples from each class. In the 10-class classification, 100 samples were selected from each class, resulting in a total of 1 000 sample images. The dataset was split, reserving 20% for testing and using the remaining portion for training.

Before modeling, the input images were resized to 256 x 256 pixels. The CNN model was constructed using a sequential model with sequential layers. To prevent overfitting and encourage generalization, we incorporated a dropout layer during training. The CNN model was applied after resizing and defining the number of classes, which included nine different disease types and one healthy class. The CNN architecture in our study was tailored to our specific requirements, involving multiple layers. The CNN model was configured with the following parameters:

- The training phase included 25 epochs.
- Each iteration involved the use of 32 images.
- Input images were resized to dimensions of 256 x 256 pixels.
- The images were labeled for classification into 2, 6, and 10 different classes.

Experiments to detect tomato leaf diseases were conducted using our model, and the steps applied in our CNN model can be observed in Figure 7.

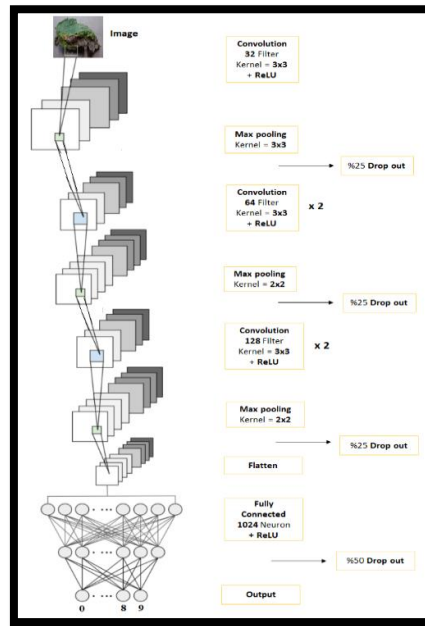


Figure 7- Our CNN Model

In our CNN model, the specified parameters were employed to conduct the following operations:

- i. ReLU-activated convolution layer with 32 filters of size 3x3.
- ii. A Maxpool layer with a 3x3 dimensional frame.
- iii. Dropout of 25% of neurons after the Maxpool layer.
- iv. ReLU-activated convolution layer with 64 filters of size 3x3.
- v. ReLU-activated convolution layer with 64 filters of size 3x3.
- vi. A Maxpool layer with a 2x2 frame.
- vii. Dropout of 25% of neurons after the Maxpool layer.
- viii. ReLU-activated convolution layer with 128 filters of size 3x3.
- ix. ReLU-activated convolution layer with 128 filters of size 3x3.
- x. A Maxpool layer with a 2x2 frame.
- xi. Dropout of 25% of neurons after the Maxpool layer.
- xii. Flattening of the data in preparation for the fully connected layer.
- xiii. ReLU-activated fully connected layer with 1024 neurons.
- xiv. Dropout of 50% of neurons after the fully connected layer.
- xv. Softmax-activated neurons in the output layer, with the number of neurons equal to the number of classes.

The CNN model employed in our study exhibited strong performance for the tomato dataset. This success can be attributed to its shorter average epoch round and higher accuracy compared to other CNN models. The key to this success lies in the experimental determination of the optimal number of layers and parameters for the CNN model, resulting in an effective and efficient model. A detailed comparison of model performances can be found in the discussion section of our study. Furthermore, the approach we used to determine the number and parameters of layers is not limited to the tomato dataset; it can be applied to other datasets as well. This eliminates dependency on a specific dataset and allows for successful application to datasets with varying numbers of classes, as demonstrated in our experimental studies.

To assess the performance of our model, the tenfold cross-validation method was used in the training and testing phase. This method involves a thorough examination of all sections of the dataset, ensuring that the model has been exposed to every sample, which contributes to a more effective learning process. Cross-validation, in particular, randomly divides the dataset into "k" groups, designating one as a test set and using the remaining groups for training. This procedure is repeated for each group, enabling the model to be trained and tested with all parts of the data, ultimately enhancing its accuracy. The cross-validation process is visualized in Figure 8.

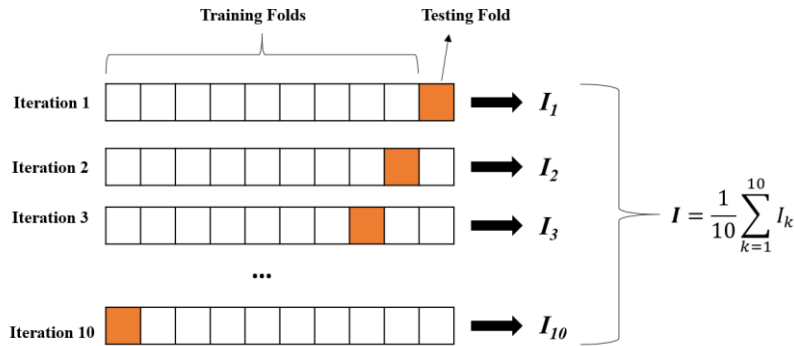


Figure 8- 10-fold cross-validation

2.7 Visualizing feature maps in our CNN model

Feature maps are the output structures generated by applying filters within a CNN. Essentially, they represent the evaluation of a particular layer. Normalizing these features aims to enhance the understanding of the detected features. In deep learning techniques, initial layers primarily identify low-level features (e.g., colors and edges), while subsequent layers identify high-level features (e.g., shapes and objects). Therefore, we incorporated feature visualization into our model. Figure 9 displays the visualization of the features of tomato leaves, while Figure 10 showcases symptom visualization. These images reveal numerous activations related to edges and textures, with a particular focus on outlining the leaf.

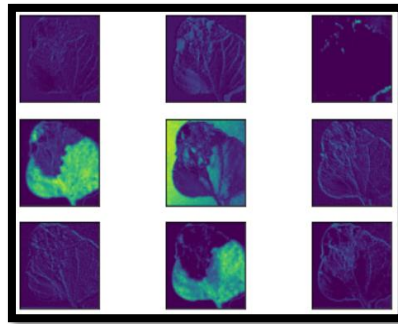


Figure 9- Feature visualization for tomato leaves

In Figure 10, the differentiation of the pixel values is seen in areas with symptoms.

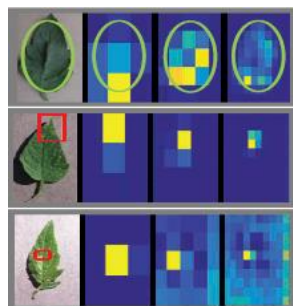


Figure 10- Symptom visualization for tomato leaves

2. Results

A comparative analysis of experimental results using various metrics is a crucial aspect to consider. To assess the effectiveness of our models, five distinct metrics were employed. The performance metrics used in our study, along with their corresponding formulas, are presented in Equations (11-15). In these equations, TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} * 100 \quad (11)$$

$$\text{Sensitivity} = \frac{\text{TP}}{(\text{TP}+\text{FN})} * 100 \quad (12)$$

$$\text{Specificity} = \frac{\text{TN}}{(\text{TN}+\text{FP})} * 100 \quad (13)$$

$$\text{PPV} = \frac{\text{TP}}{(\text{TP}+\text{FP})} * 100 \quad (14)$$

$$\text{NPV} = \frac{\text{TN}}{(\text{TN}+\text{FN})} * 100 \quad (15)$$

PPV and NPV serve as indicators of a test's clinical significance. Sensitivity signifies the percentage of true positives (e.g., 95% sensitivity means that 95% of individuals with the targeted disease will test positive), while specificity indicates the percentage of true negatives (e.g., 95% specificity suggests that 95% of individuals without the targeted disease will test negative). Accuracy is a measure of correctly identifying both diseased and healthy datasets. Tables 5, 6, and 7 present the classification results for conventional learning models in our study. Table 5 showcases our model's ability to distinguish between diseased and healthy tomato leaf images. A closer examination of the table reveals that the SVM method attained the highest accuracy rate, reaching 92.5%. Furthermore, the kNN method, a simple yet effective approach, achieved the highest sensitivity rate of 98%, as depicted in the same table.

Table 5- Binary class classification performance metrics with classical learning methods (the most successful method for each metric is shown in bold)

Methods	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)	NPV (%)
ELM	91.80	95.80	87.80	88.70	95.40
SVM	92.50	96.00	89.00	89.70	95.70
kNN	89.00	98.00	80.00	83.10	97.60

Table 6 illustrates the performance metrics when six classes are created by merging the five most frequent diseases with the healthy class. Similar to binary classification, SVM stands out as the most effective method. However, after the six-class classification, it appears that the outcomes of all methods are quite similar.

Table 6- 6-Class classification performance metrics with classical learning methods (the most successful method for each metric is shown in bold)

Methods	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)	NPV (%)
ELM	81.30	95.90	96.90	84.50	99.30
SVM	83.00	94.00	98.60	92.10	98.90
kNN	82.30	97.30	96.40	82.90	99.50

Table 7 presents the results obtained by incorporating all classes in the dataset. In the table, success decreases as the number of classes increases. The results of the model with 10 different classes indicate that the SVM method achieved a higher accuracy rate compared to other methods.

Table 7- 10-Class classification performance metrics with classical learning methods (the most successful method for each metric is shown in bold)

Methods	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)	NPV (%)
ELM	78.70	94.10	97.40	78.90	99.40
SVM	81.40	92.70	98.90	89.40	99.20
kNN	79.60	95.03	97.80	81.40	99.50

When the results obtained from Tables 5, 6, and 7 are analyzed, it becomes evident that the kNN method excels in terms of successfully detecting diseased samples, particularly for sensitivity. On the other hand, the SVM method proved to be the most effective in distinguishing healthy images, emphasizing its specificity.

Table 8 presents the implementation times of the methods utilized in our study. The table illustrates how the number of classes impacts the implementation time of these methods. Binary classification emerges as the quickest method, while the ELM method had the fastest performance when classifying all 10 classes. Notably, the SVM method, despite providing the highest accuracy rates, exhibited slower performance when classifying 6 and 10 classes.

Table 8- Average time of classification for methods

Methods	Binary	6-Class	10-Class
ELM	12.90	39.80	115.90
SVM	5.60	235.20	1938.60
kNN	2.30	45.90	470.90
CNN	265.00	432.00	505.00

As depicted in Table 9, CNN exhibited the highest accuracy rates for binary, 6, and 10 classes, but it required a longer training duration compared to classical learning methods. Interestingly, binary classification also proved to be the quickest method within the CNN framework. What's particularly intriguing is that our proposed CNN architecture displayed less sensitivity to the number of classes when contrasted with ELM, SVM, and kNN methods, where an increase in the number of classes substantially augmented execution time. For instance, in SVM, the execution time difference between binary classification and 10-class classification was approximately 345 times, while in the CNN method, this ratio was only about 2 times. This observation implies that our proposed CNN method is more versatile and original.

The classification results obtained with our deep learning models are presented in Table 9, revealing that all accuracy results are quite similar. Furthermore, as the number of classes increases, accuracy tends to decrease.

Table 9- Classification performance metrics according to different class numbers with deep learning methods (the most successful method for each metric is shown in bold)

Number of classes	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)	NPV (%)
Binary Class	99.50	100.00	98.97	99.04	100.00
6-Class	98.50	98.50	99.70	98.50	99.70
10-Class	97.00	97.00	99.67	97.00	99.67

In our study, modeling was conducted for three different class numbers: binary, 6-class, and 10-class. Table 10 compares the accuracy values of the CNN architecture with those of classical learning methods. Upon reviewing the table, it is clear that the original CNN model yielded considerably more successful results than classical learning methods. This can be attributed to the fact that the parameters and layers employed in the CNN model were determined through extensive experimental research, a fact reflected in Table 10.

Table 10- Accuracy values according to different class numbers with deep and classical learning methods (the most successful method for different numbers of classes is shown in bold)

Number of classes	CNN (%)	ELM (%)	SVM (%)	kNN (%)
Binary Class	99.50	91.80	92.50	89.00
6-Class	98.50	81.30	83.00	82.30
10-Class	97.00	78.70	81.40	79.60

Figures 11, 12, and 13 illustrate the current state of the dataset and the number of correct and incorrect classifications made by the model. The vertical axis represents the actual values, while the horizontal axis represents the predicted values. The confusion matrix is utilized to determine the number of TP, TN, FP, and FN. TP values are displayed as numerical values on the diagonal, signifying highly accurate disease classification.

Upon analyzing the confusion matrix, it can be concluded that each class exhibits the highest rate of correct estimations. More specifically, according to Figure 11, the highest correct estimation is for the healthy state, while the lowest correct estimation is for leaf mold disease (Figure 12) and yellow leaf curl virus disease (Figure 13). Furthermore, when inspecting the incorrect predictions, it is apparent that bacterial spot, target spot, and yellow leaf curl virus are diseases that are most frequently confused with each other. To mitigate this confusion between these diseases, it may be beneficial to analyze mixed features of the images and implement certain preprocessing steps to enhance the accuracy of classification.

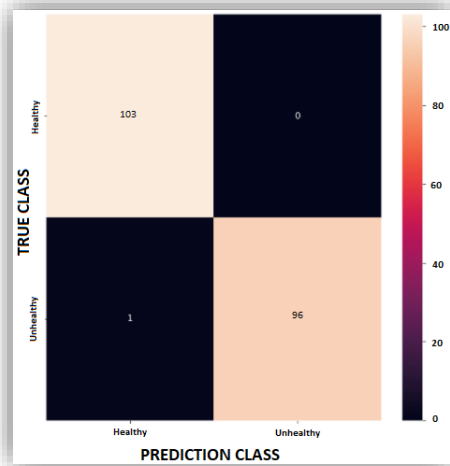


Figure 11- Confusion Matrix for Binary Class

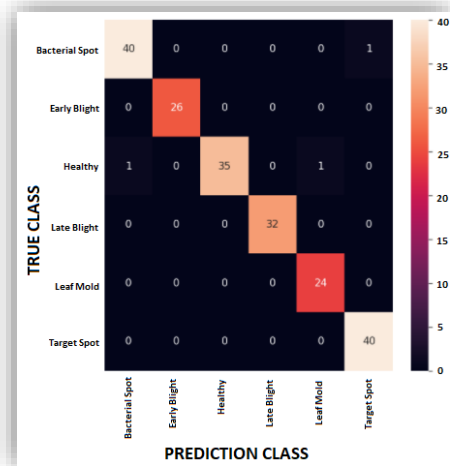


Figure 12- Confusion Matrix for 6-Class

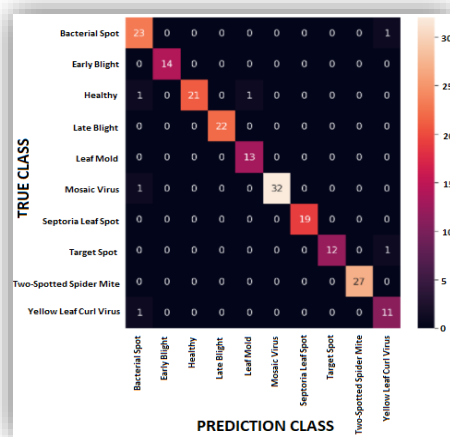


Figure 13- Confusion Matrix for 10-Class

3. Discussion

Our study aimed to develop a novel CNN framework for the automatic classification of tomato leaf images (Lamrahi 2021). We conducted a comparative analysis of the CNN method with models created using different classical learning methods, as demonstrated in Tables 5, 6, 7, and 9, which present the performance metrics of these models. The results consistently indicate that the CNN method outperformed the classical models across all class numbers. This comparison emphasizes the advantages

of deep learning models over classical machine learning methods, particularly their capacity to extract features from raw data without the need for expert knowledge.

The findings of our study hold significant implications for the field of agriculture. Traditionally, disease detection in crops relies on manual observations by farmers, which can be time-consuming and costly, especially when diseases are detected late. Our method can be applied to process images captured from vast agricultural areas using drone-like devices, enabling the early detection of plant diseases and facilitating timely interventions to prevent yield loss. Table 11 provides a comparison of our study with other studies related to the tomato dataset. It is worth noting that the studies (Kapucuoglu 2011; Anonymous 2021b; c) listed in the table are not academic studies, as the dataset was not used in an academic context; they were sourced from Kaggle for comparison purposes.

Table 11- Comparison table of current studies of the tomato (Lamrahi 2021) dataset

<i>Study</i>	<i>Model</i>	<i>Number of Classes</i>	<i>Loss (%)</i>	<i>Accuracy (%)</i>	<i>Average Epoch Tour</i>
(Anonymous 2021b)	Sequential	10	14.20	95.35	308
(Anonymous 2021c)	VGG-16	10	1.84	97.96	314
(Kapucuoglu 2011)	AlexNet	10	8.92	97.20	233
Our Study	Sequential	2	0.04	99.50	11
Our Study	Sequential	6	2.13	98.50	17
Our Study	Sequential	10	9.33	97.00	20

As indicated in Table 11, our study achieved the lowest loss rate, the shortest average epoch count, and the highest accuracy for 2-class classification. In comparison to the study (Kapucuoglu 2011) which utilized AlexNet for 10-class classification and achieved an accuracy rate of 97.20%, our model attained an accuracy of 97% with a shorter average epoch count when subjected to 10-fold cross-validation. Furthermore, when our method is applied to classify the 5 most common diseases and one health status (6-class), we obtained favorable results. The proposed approach and CNN model consistently demonstrated high success across all classification methods while requiring fewer epochs than other studies, underscoring the effectiveness and robustness of our method.

4. Conclusions

The results demonstrate that the deep learning model, particularly the CNN architecture, outperforms classical methods in terms of both accuracy and efficiency. The CNN model consistently achieved high accuracy rates for all the different classification methods employed in the study, including the identification of multiple diseases and health status. Furthermore, the CNN model's ability to automatically extract features from raw data without requiring expert input is a notable advantage over classical feature extraction methods, which often rely on domain-specific knowledge for dataset-specific feature selection.

This study has significant implications for the agricultural sector, as early disease detection in plants is crucial for improving productivity and reducing costs. Digital detection tools developed through smart agriculture studies can facilitate early diagnosis and treatment of plant diseases, ultimately leading to higher crop yields and higher-quality agricultural products. Overall, this study underscores the potential of deep learning methods, particularly CNN models, in the realm of plant disease detection, and emphasizes the importance of ongoing research in this field.

Tomato plants are susceptible to various types of diseases. As a result, our study initially focused on distinguishing between diseased and healthy leaves and subsequently categorizing the diseased leaves into 6 or 10 different types. The results reveal that our original CNN model achieved the highest success rates for 2, 6, and 10-class classifications, with accuracy rates of 99.5%, 98.50%, and 97.0%, respectively. In comparison to classical methods, the CNN network we designed consistently delivered significantly superior results. Classical methods typically involve data preprocessing, while CNN methods can directly extract features from raw data, eliminating the need for additional feature extraction techniques. This attribute significantly contributes to the success of our model. Our findings underscore the importance of computer-aided recognition and detection systems for enhancing agricultural productivity. Additionally, the independence of our model from the dataset enables its application to different plant species and various disease types, making it a valuable contribution to the field of agriculture.

In future studies, we plan to develop a system capable of detecting various plant species and disease types, while also assessing disease severity based on images of afflicted plants. Our goal is to make this system compatible with mobile devices, further supporting the agricultural industry by enhancing production and efficiency.

Funding

The authors did not receive support from any organization for the submitted work.

Data Availability

The present work utilizes one dataset which is tomato data from Kaggle developed by Nouaman Lamrahi. This dataset is cited in the manuscript in Lamrahi (2021).

Conflict of interest

The authors declare that there is no conflict of interest.

References

- Adebayo S E, Hashim N, Abdan K & Hanafi M (2016). Application and potential of backscattering imaging techniques in agricultural and food processing—A review. *Journal of Food Engineering*, 169: 155–164
- Altuntaş Y & Kocamaz F (2021). Deep feature extraction for detection of tomato plant diseases and pests based on leaf images. *Celal Bayar University Journal of Science*, 17(2): 145–157
- Anonymous (2021a). *Complex Projective 4-Space*. <https://cp4space.wordpress.com/page/3/> [2021-06-10]
- Anonymous (2021b). *Tomato Plant Disease Detection by RAVI*. <https://www.kaggle.com/ravibalas1999/tomotoplant-disease-detection> [2021-07-02]
- Anonymous (2021c). *VGG16 by MMDRAGE*. <https://www.kaggle.com/mmdrage/vgg16-fine-tuning-and-98-55-val-acc> [2021-07-02]
- Arakeri M P (2016). Computer vision based fruit grading system for quality evaluation of tomato in agriculture industry. *Procedia Computer Science*, 79: 426–433
- Arya S, Mount D M, Netanyahu N S, Silverman R & Wu A Y (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45 (6): 891–923
- Bhandari M, Shahi T B, Neupane A & Walsh K B (2023). BotanicX-AI: Identification of Tomato Leaf Diseases Using an Explanation-Driven Deep-Learning Model. *Journal of Imaging*, 9 (2): 53. <https://doi.org/10.3390/jimaging9020053>
- Brahimi M, Boukhalfa K & Moussaoui A (2017). Deep learning for tomato diseases: classification and symptoms visualization. *Applied Artificial Intelligence*, 31 (4): 299–315
- Burgos-Artizzu X P, Ribeiro A, Tellaeche A, Pajares G & Fernández-Quintanilla C (2010). Analysis of natural images processing for the extraction of agricultural elements. *Image and Vision Computing*, 28 (1): 138–149
- Chen X, Zhou G, Chen A, Yi J, Zhang W & Hu Y (2020). Identification of tomato leaf diseases based on combination of ABCK-BWTR and B-ARNet. *Computers and Electronics in Agriculture*, 178: 105730
- Durmuş H, Güneş E O & Kırıcı M (2017). Disease detection on the leaves of the tomato plants by using deep learning. In: Proceedings of the 6th International Conference on Agro-Geoinformatics (IEEE), 07-10 August, Fairfax VA, USA, pp. 1–5
- Dutta M K, Sengar N, Minhas N, Sarkar B, Goon A & Banerjee K (2016). Image processing based classification of grapes after pesticide exposure. *LWT-Food Science and Technology*, 72: 368–376
- Gerdan D, Koç C & Vatandaş M (2023). Diagnosis of Tomato Plant Diseases Using Pre-Trained Architectures and A Proposed Convolutional Neural Network Model. *Journal of Agricultural Sciences (Tarım Bilimleri Dergisi)*, 29(2): 618-629 <https://doi.org/10.15832/ankutbd.957265>
- Gonzalez-Huitron V, León-Borges J A, Rodriguez-Mata A E, Amabilis-Sosa L E, Ramírez-Pereda B & Rodriguez H (2021). Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4. *Computers and Electronics in Agriculture*, 181: 105951
- Huang G B, Zhou H, Ding X & Zhang R (2011). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42 (2): 513–529
- Huang G B, Zhu Q Y & Siew C K (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of the IEEE International Joint Conference on Neural Networks, 25-29 July, Budapest, Hungary, pp. 985–990
- Huang G B, Zhu Q Y & Siew C K (2006). Extreme learning machine: theory and applications. *Neurocomputing*, 70 (1–3): 489–501
- Jiang F, Lu Y, Chen Y, Cai D & Li G (2020). Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Computers and Electronics in Agriculture*, 179: 105824
- Kapucuoglu K (2011). *Plant Diseases Classification Using AlexNet*. <https://www.kaggle.com/koksal1994/plant-diseases-classification-using-alexnet> [2021-08-03]
- Karthik R, Hariharan M, Anand S, Mathikshara P, Johnson A & Menaka R (2020). Attention embedded residual CNN for disease detection in tomato leaves. *Applied Soft Computing*, 86: 105933
- Kuta Ł, Li Z, Stopa R, Komarnicki P & Ślupska M (2020). The influence of manual harvesting on the quality of picked apples and the Picker's muscle load. *Computers and Electronics in Agriculture*, 175: 105511
- Lamrahi N (2021). *Tomato Dataset*. <https://www.kaggle.com/noulam/tomato> [2021-04-01]
- Lim J S & Oppenheim A V (1979). Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*, 67 (12): 1586–1604
- Liming X & Yanchao Z (2010). Automated strawberry grading system based on image processing. *Computers and electronics in agriculture*, 71: S32–S39
- Methun N R, Yasmin R, Begum N, Rajbongshi A & Islam M E (2021). Carrot disease recognition using deep learning approach for sustainable agriculture. *International Journal of Advanced Computer Science and Applications*, 12 (9): 732-741
- Ojala T, Pietikäinen M & Mäenpää T (2000). Gray scale and rotation invariant texture classification with local binary patterns, *Computer Vision - ECCV 2000 (Springer)*, 1842: 404–420
- Ouhami M, Es-Saad Y, Hajji M El, Hafiane A, Canals R & Yassa M El (2020). Deep transfer learning models for tomato disease detection, In: Proceedings of the Image and Signal Processing: 9th International Conference, ICISP 2020, Marrakesh, Morocco, 2020, Proceedings 9 (pp. 65-73). Springer International Publishing.2020. 65–73. Springer
- Park K, ki Hong Y, hwan Kim G, & Lee J (2018). Classification of apple leaf conditions in hyper-spectral images for diagnosis of Marssonina blotch using mRMR and deep neural network. *Comput. Electron. Agric.*, 148: 179–187

- Rahman S U, Alam F, Ahmad N & Arshad S (2023). Image processing based system for the detection, identification and treatment of tomato leaf diseases. *Multimedia Tools and Applications*, 82 (6): 9431–9445. <https://doi.org/10.1007/s11042-022-13715-0>
- Rehman Z U, Khan M A, Ahmed F, Damaševičius R, Naqvi S R, Nisar W & Javed K (2021). Recognizing apple leaf diseases using a novel parallel real-time processing framework based on MASK RCNN and transfer learning: An application for smart agriculture. *IET Image Processing*, 15 (10): 2157–2168
- Schölkopf B & Smola A J (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.
- Sembiring A, Away Y, Arnia F & Muharar R (2021). Development of concise convolutional neural network for tomato plant disease classification based on leaf images. *Journal of Physics: Conference Series IOP Publishing*. 1845(1): 012009
- Sethy P K, Barpanda N K, Rath A K & Behera S K (2020). Deep feature based rice leaf disease identification using support vector machine. *Computers and Electronics in Agriculture*, 175: 105527
- Temniranrat P, Kiratiratanapruk K, Kitvimonrat A, Sinthupinyo W & Patarapuwadol S (2021). A system for automatic rice disease detection from rice paddy images serviced via a Chatbot. *Computers and Electronics in Agriculture*, 185: 106156
- Tian K, Li J, Zeng J, Evans A & Zhang L (2019). Segmentation of tomato leaf images based on adaptive clustering number of K-means algorithm. *Computers and Electronics in Agriculture*, 165: 104962
- Tian K, Zeng J, Song T, Li Z, Evans A & Li J (2022). Tomato leaf diseases recognition based on deep convolutional neural networks. *Journal of Agricultural Engineering*, 54(1). <https://doi.org/10.4081/jae.2022.1432>
- Tm P, Pranathi A, SaiAshritha K, Chittaragi N B & Koolagudi S G (2018). Tomato leaf disease detection using convolutional neural networks, In: Proceedings of the Eleventh International Conference on Contemporary Computing (IC3) pp. 1–5
- Wspanialy P & Moussa M (2020). A detection and severity estimation system for generic diseases of tomato greenhouse plants. *Computers and Electronics in Agriculture*, 178: 105701
- Xu P, Wu G, Guo Y, Yang H & Zhang R (2017). Automatic wheat leaf rust detection and grading diagnosis via embedded image processing system. *Procedia Computer Science*, 107: 836–841
- Zhong Y & Zhao M (2020). Research on deep learning in apple leaf disease recognition. *Computers and Electronics in Agriculture*, 168: 105146



Copyright © 2024 The Author(s). This is an open-access article published by Faculty of Agriculture, Ankara University under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and reproduction in any medium or format, provided the original work is properly cited.