EURASIP Journal on Image and Video Processing

RESEARCH

Open Access

An optimized capsule neural networks for tomato leaf disease classification

Lobna M. Abouelmagd[1], Mahmoud Y. Shams[2*], Hanaa Salem Marie[3] and Aboul Ella Hassanien[4]

*Correspondence:
mahmoud.yasin@ai.kfs.edu.eg

[1] Misr Higher Institute for Commerce and Computers, Mansoura, Egypt
[2] Faculty of Artificial Intelligence, Kafrelsheikh University, Kafr El-Sheikh 33516, Egypt
[3] Faculty of Artificial Intelligence, Delta University for Science and Technology, Gamasa 35712, Egypt
[4] Faculty of Computers and Artificial Intelligence, Cairo University, Cairo, Egypt

Abstract

Plant diseases have a significant impact on leaves, with each disease exhibiting specific spots characterized by unique colors and locations. Therefore, it is crucial to develop a method for detecting these diseases based on spot shape, color, and location within the leaves. While Convolutional Neural Networks (CNNs) have been widely used in deep learning applications, they suffer from limitations in capturing relative spatial and orientation relationships. This paper presents a computer vision methodology that utilizes an optimized capsule neural network (CapsNet) to detect and classify ten tomato leaf diseases using standard dataset images. To mitigate overfitting, data augmentation, and preprocessing techniques were employed during the training phase. CapsNet was chosen over CNNs due to its superior ability to capture spatial positioning within the image. The proposed CapsNet approach achieved an accuracy of 96.39% with minimal loss, relying on a 0.00001 Adam optimizer. By comparing the results with existing state-of-the-art approaches, the study demonstrates the effectiveness of CapsNet in accurately identifying and classifying tomato leaf diseases based on spot shape, color, and location. The findings highlight the potential of CapsNet as an alternative to CNNs for improving disease detection and classification in plant pathology research.

Highlights

- This paper is interested in tracking plant diseases such as leaf disease in Tomatoes, based on optimized Deep Learning Capsule Net.
- Early detection of the most common diseases based on leaf images can help us track and enhance food quality.
- Capsule Networks is an optimized alternative method more efficient than traditional CNN for detecting all leaf images and determining all locations. However, it tackles the shortage of conventional CNN.

Keywords: Drone, Tomato leaf diseases, Convolutional Neural Networks, Capsule Neural Networks, Adam optimizer

1 Introduction

The detection of plant diseases using machine learning techniques has gained significant attention due to the limitations and biases associated with human observation [1]. Unmanned aerial vehicles (UAVs) have been widely employed in precision agriculture for crop monitoring and disease surveillance, leveraging their ability to capture extensive data and information. In particular, the tomato crop holds strategic importance, necessitating early detection and control of diseases. Leaves play a crucial role in detecting plant diseases, as each disease manifests in specific spots characterized by unique colors and orientations. Therefore, there is a need to develop a robust method that can detect diseases by analyzing the shape, color, and spatial positioning of these spots within the tomato plant leaves [2–4]. Crop monitoring, biomass estimate, field mapping, plant population counts, weed management, and spraying have all been proven to benefit from the use of unmanned aerial vehicles (UAVs) for thorough surveys in precision agriculture. Disease detection and surveillance with a UAV is becoming more common as cameras, sensors, motors, rotors, controllers, and other components collaborate to capture a vast amount of data and information to improve agricultural techniques [5]. Capturing images is one of the most basic functions of a UAV [6, 7]. In recent years, researchers have increasingly employed a combination of image processing, denoising, and deep learning techniques to extract valuable information from collected plant leaf image [3, 8, 9].

The tomato crop is considered a strategic crop for many countries worldwide, so attention has been paid to the early detection and control of diseases. Leaves can detect plant diseases of all kinds because it dramatically affects the whole plant. Recent works stated that every disease in a given spot has a color associated with another color and in a specific direction. Therefore, it is essential to find a way to detect the disease that affects plants through the shape, color, and orientation of the spots inside the leaf, for enhancing food nutrition schemes [10–15].

In the early years, image processing with data mining techniques was used to recognize plant diseases [16–18]. The major approaches used in this technique were K-nearest neighbor (KNN), backpropagation neural network, support vector machine (SVM), and spatial gray-level dependency matrices. These approaches effectively recognize healthy and unhealthy plant leaves [19–22]. The features of the image can be extracted using image processing techniques such as color analysis and thresholding approaches [23]. Furthermore, deep learning (DL) algorithms are used to extract features in an automated way. Researchers have used them to extract distinctive features in plant disease recognition with the least amount of knowledge and less human effort. These algorithms consist of interconnected architectural layers through which data are rep-resented, and higher-level features can be extracted from the last layers of networks. However, the low-level features are extracted from the lower layers [24, 25].

A convolutional neural network (CNN) is a popular and widely used DL technique for classification and feature extraction in various applications such as natural language processing, speech processing, and computer vision [26–28]. A CNN simulates a complex series of cells in a cat's visual cortex. Parameter sharing, sparse connections, and equivalent representations are CNN's three key benefits, and they simulate complex series of cells in a cat's visual cortex. However, the traditional method uses fully connected networks, whereas CNN uses local connections and shared weights to use the two-dimensional input image. This method yields a network with considerably fewer parameters, making it simpler and easier to practice. This process is like that which occurs in visual cortex cells. Small parts of a scene, rather than the entire scene, are vulnerable to these cells. In other words, the cells serve as local filters over the information, extracting spatially using local correlation from the data [29–33].

There are many convolutional layers in a typical CNN, followed by pooling (subsampling) layers, and finally, fully connected layers in the final stage [32]. Although several flaws have been noted, CNN pre-trained architectures have been successfully applied to plant disease detection [34, 35]. A CNN does not place a high value on the orientation and spatial relationships between image components; therefore, it does not consider the spatial relationship between its features. However, it is unresistant to affine transformations, necessitating a large amount of data for learning, including all possible image orientations. Nevertheless, constructing a model requires a large dataset. This results in longer training times and overuse of computing resources. A CNN is called invariant rather than equal-variant because of its pooling operation [36].

It does not capture the relative spatial and orientation relationships, and it is easily tricked by image orientation or a shift in a pose. In a CNN, the max-pooling layer is essential to reduce the spatial information of the data transferred to the next layer by down sampling it. This process can be termed a downside for a CNN because it cannot propagate spatial hierarchies between different objects [37].

In-plant diseases, the leaf is the first part affected. Researchers discovered that every disease has certain spots, and each Spot has a color associated with another color and a specific leaf location. Therefore, it is essential to find a way to detect the disease that affects plants through the shape of their spots, col-or, and destination inside the leaf [38]. Because traditional CNN techniques cannot detect diseases that depend on spots' location on plants, this study will not consider them. It is practical if experiments relating to this study are conducted outside the laboratory [39].

Lately, an unprecedented type of neural network that suggests a "capsule" concept [40]. In this way, the capsule must encode a particular visual feature's presence, and its transformations can be subject to a specific application or domain. This paper defines the capsule as a set of neurons representing the entire feature and parameters related to the instantiation of the features. Although a traditional CNN uses kernels' scalar activations, capsule vectors enrich the network with information. Adam is a stochastic optimization approach that just requires first-order gradients and uses little memory. This method uses estimations of the first and second moments of the gradients to calculate individual adaptive learning rates for distinct parameters; the name Adam comes from Adaptive Moment Estimation. It helps adjust the learning rate for each neural network weight using the estimates of the first and second moments of the gradient [41].

The tomato crop is one of the important crops that we use daily, making it indispensable. Early detection of diseases in plants is one of the leading practices that agricultural engineers use to limit the spread of the disease, and it can be achieved most times through plant leaves. The tomato leaf is one of the necessary things that express the health condition and the degree of ripeness of the fruit. There are two forms of tomato fruits: healthy and unhealthy. It is challenging to stir the fruit in three-dimensional directions to distinguish the external shape to detect diseases in the ripe and immature fruits. Therefore, early detection of the disease through their tree leaves is needed [42, 43].

Presently, researchers have made several attempts to use artificial intelligence to predict, diagnose, and investigate suitable ways for plant disease classification and to take action after detecting the symptoms of leaf diseases. Sardogan et al. [44] presented the CNN model and Learning Vector Quantization (LVQ) algorithm-based method to classify tomato leaf disease. They used 400 leaf images for training and 100 leaf images for testing. The average accuracy of 86% was obtained for five class labels of leaf diseases, including healthy, bacterial Spot, late Blight, septoria spot, and yellow curved.

Mokhtar et al. [45] used SVM with different kernel functions to classify two different yellow leaf curl diseases for 200 infected tomato leaf images, and 90% average accuracy was obtained. Besides the five mentioned classes obtained in [44, 46], a CNN to classify six class leaf diseases, adding spider mites to their study. The average accuracy achieved was 76% for 600 input images (100 for each class) presented by [46]. A large-scale dataset consisting of 14,828 images of tomato leaves infected with nine diseases, which are 4032, yellow leaf curl virus; 325, mosaic virus; 1356, target spot; 1628, spider mites; 904, leaf mold; 1723, septoria spot; 1781, late Blight; 952, early blight; and 2127, bacterial Spot, was presented by Brahimi et al. [47] using a CNN. They compared the CNN model with shallow models and hand-crafted features. The accuracy values obtained were 94.54% and 95.46% for the SVM and random forest shallow models. Meanwhile, 98.66% and 97.35% were obtained for Alexnet, and 99.18% and 97.71% were obtained when GoogleNet was used with and without pretraining.

Nine class labels were further presented by Gao et al. [8] to identify leaf disease in tomatoes using a CNN. They also used the applied transfer learning algorithms such as Res-net, Alexnet, and GoogleNet with stochastic gradient descent and Adam optimizers. They achieved significant accuracy of 96.51% (Resnet). Conditional Generative Adversarial Network (CGAN) with DenseNet121 was presented by Abbas et al. [48] to generate synthetic images of tomato plant leaves based on ten class labels of the PlantVillage dataset that contains 16,012 images with a class-wise tomato image distribution. They achieved 99.17% accuracy compared with 99.51% and 98.65% for five and seven class labels. Atila et al. [49] present an EfficientNet deep learning model to classify thirty-nine categories of plant diseases, including tomato disease, with average accuracy 99.91%.

Chowdhury et al. [50] presented a CNN EffecientNet to classify tomato leaf diseases with three different scenarios. The first scenario uses two class labels that classify tomato into two healthy and non-healthy. The second scenario is based on segmented six class labels. Finally, the third scenario is the utilization of ten class labels, and they used Adam optimizer for both segmentation and classification model. They achieved an average accuracy of 99.00% based on tomato leaf images. Furthermore, Tan et al. [51] presented a comparative study between the traditional Machine Learning approaches and Deep Learning approaches to classify tomato diseases using leaf images extracted from the PlantVillage dataset. The dataset contains ten categories such that the number of healthy leaf images is 1591, and the nine remaining infected leaf images are varied from 373 to 5357. They obtained 82.10%, 91.00%, and 82.70% using the classical K-Nearest Neighborhood (KNN), Support Vector Machine (SVM), and Random Forest (RF), respectively. While they obtained 92.70%, 98.90%, 99.70%, 98.90%, and 91.20% using AlexNet, VGG16, ResNet34, EffecientNet, and MobileNetV2, DL approaches respectively.

Although their study results were promising, a track change of leaf color or components is essentially required. However, our proposed architecture can handle this problem using CapsNet because it tackles the disadvantage of traditional CNN, tracks the variably of leaf changes, and enhances the identification process's performance. Therefore, we recommend using a capsule network that can encode the visual feature and a set of neurons representing the entire feature and realize the parameters related to the instantiation of the said feature. The novelty of this research lies in several key aspects, which significantly differentiate it from existing approaches and contribute to its significance in the field of plant disease detection. Firstly, while previous studies have focused on using traditional CNN architectures for plant disease classification, this research introduces the utilization of an optimized capsule neural network (CapsNet). CapsNet addresses the limitations of CNNs by capturing relative spatial and orientation relationships, which are crucial for accurate disease detection. By leveraging the unique properties of CapsNet, such as encoding visual features and utilizing a set of neurons representing the entire feature, the proposed architecture improves the performance of disease identification. Secondly, the optimization of CapsNet's hyperparameters using the Adam optimizer is another novel contribution. The Adam optimizer is known for its ability to adaptively adjust the learning rate for each weight in the neural network. This optimization technique improves the efficiency and effectiveness of the CapsNet model, leading to enhanced disease classification accuracy. Compared to existing approaches, the proposed framework offers several significant advantages. The use of CapsNet addresses the shortcomings of traditional CNNs, allowing for improved detection and classification of tomato leaf diseases based on spot shape, color, and orientation. The integration of drone-captured images enhances the quality and resolution of the input data, facilitating more precise disease analysis. Additionally, the optimization of CapsNet's hyperparameters using the Adam optimizer further boosts the model's performance. This paper highlights the following contributions:

1. The development of a computer vision system for the accurate classification of tomato leaf diseases.
2. The construction of a CapsNet-based architecture that overcomes the limitations of traditional CNNs and efficiently extracts and classifies plant images.
3. The optimization of CapsNet hyperparameters using the Adam optimizer.
4. The optimized CapsNet model offers farmers a reliable tool for timely disease identification and management, contributing to improved agricultural productivity and food security.
5. The successful distinction of ten types of tomato leaf diseases based on the color, shape, and orientation of the spots within the leaves.

The rest of this paper is organized as follows. Section 2 demonstrates that the proposed methodologies include the proposed architectures. Section 3 includes the evaluation of the experimental results. The conclusion and future work are presented in Sect. 4.

2 Methodologies

This paper presents a computer vision system using CapsNet to overcome traditional CNN's limitations and improve the identification process's efficiency. Figure 1 depicts a block diagram of the proposed CapsNet architecture. CapsNet comprises a group of neurons that perform a significant internal computation before compressing the results into smaller vectors of highly informative outputs. When installed in a system's brain, this network is inspired by a Mini column in which each capsule learns to identify an obliquely distinct visual entity across a limited set of viewing conditions and deformations [52, 53].

Each capsule's detection probability encodes the feature as the length of its output vector. The detected feature is guided to the parameters for instantiation. Consequently, the feature transfers around the picture or affects its state somehow. The likelihood remains constant (the length of the vector does not change), but the vector's orientation changes. When an object "moves over the manifold of potential appearances" in the image, Hinton refers to this movement as "activities equivariant." Simultaneously, the detection probabilities remain constant, which is the type of invariance that we can aim for rather than the type given by CNNs with max pooling [54].

To ensure that the vector length or the likelihood of an object remains between zero and one, the nonlinear function called "squashing" was used in Eq. (1) to keep both the length and direction of the input vector in the (0, 1) range.

$$\mathrm{R_j} = \frac{||\mathrm{p_j}||^2}{1 + ||\mathrm{p_j}||^2} \frac{\mathrm{p_j}}{\left|\left|\mathrm{p_j}\right|\right|} \tag{1}$$

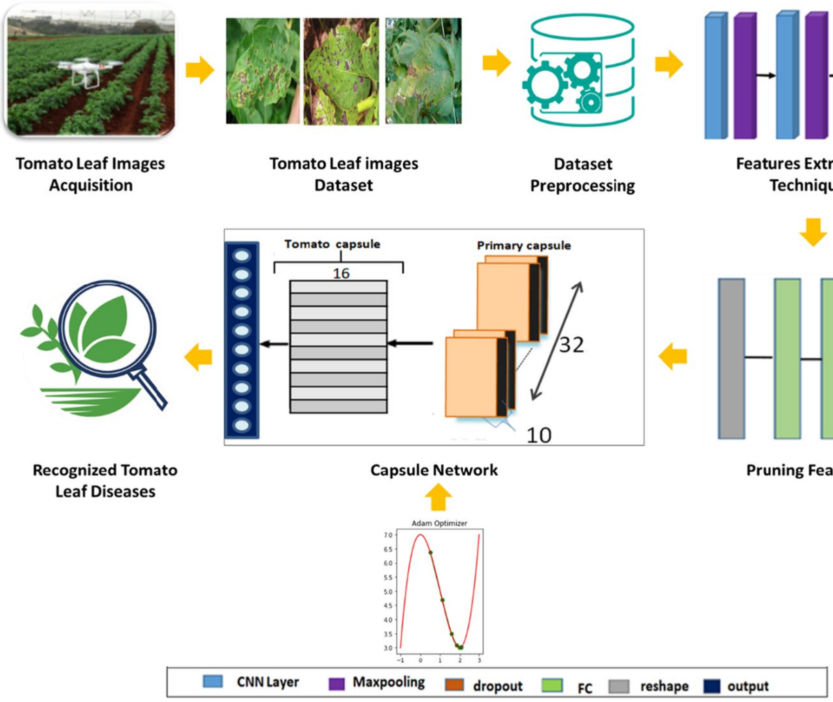


Fig. 1 Proposed capsule network architecture for leaf classification

where $\mathrm{R_j}$ represents the vector output and capsules j and $\mathrm{p_j}$ represent the input.

The total input of capsule $\mathrm{P_j}$ is a weighted sum over all "predictive vectors," $\hat{o}_{\mathrm{j|i}}$. The capsules in the previous layer were calculated by multiplying the weight matrix $\mathrm{W_{ij}}$ by the output $\mathrm{o_i}$ of a capsule in the layer below.

$$p_{\mathrm{j}} = \sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{c_{ij}}\hat{\mathrm{o}}_{\mathrm{j|i}} \tag{2}$$

$$\hat{\mathrm{o}}_{\mathrm{j|i}} = \mathrm{W_{ij}}\mathrm{o_i} \tag{3}$$

where cij denotes "coupling coefficients" obtained from the dynamic routing algorithm, the "coupling coefficients" between the i capsule and all capsules in the layer above the sum to 1 was determined by the "routing softmax," representing the initial log_bits bij of the prior probabilities of that capsule that should be coupled to capsule j.

$$\mathrm{c_{ij}} = \frac{\exp\left(\mathrm{b_{ij}}\right)}{\sum_{\mathrm{k}} \exp\left(\mathrm{b_{ij}}\right)} \tag{4}$$

As investigated in Fig. 1, the proposed architecture to classify tomato leaf diseases based on optimized CapsNet consists of the following stages: Acquiring a dataset of tomato leaf images by drone is in its early stages.

- Dataset preprocessing.
- Feature extraction.
- Pruning feature.
- CapsNet and optimization.

In this paper, CNN, the convolution layer, holds an essential function to detect the features from an image pixel. Deeper CNN layers detect simple features like edges and color. The use of the max-pooling layer was limited. It did not significantly affect the spatial relationship of the features. Still, using CNN with max-pooling benefited the model as it reinforced other features such as colors and borders. We have hybrid CNN convolutional layers with Capsule network layers in the proposed architecture. The hybrid removes the limitation of each disadvantage. The CNN requires downsampling by max-pooling layer, and the Capsule network is equivariant that increases opportunity in classification rate with the reduced number of nodes per feature vector. The architecture includes three convolutional layers for feature extraction and interoperates three max-pooling for reducing the oversize of feature vectors as input to the Capsule network [55]. It is specialized in determining the rotation, equivariant eventually it is complex and computationally cost. In turn, the principal purpose and rationality of the network basis on the hybrid of feature extraction using the limited number of the traditional CNN method in support of the Capsule network having equivariant without complexity in cost and time [56, 57].

2.1 The acquisition of a dataset stage

The source of the experimental data is the standard dataset founded in "Sharma, S. R. 2020. Plant diseases. Available at https://www.kaggle.com/saroz014/plant-diseases (Last Accessed on 18 Sep. 2022). The data are not collected from UAV as we present a perception of the utilization of the UAV drone to take an accurate picture of the tomato leaves in the future. The image gathered from UAV has complex background than the image taken in the laboratory. In the actual workflow, the stated scenario is the proposed framework of the current is being designed UAV. The workflow consists of an edge node responsible for transforming the real-world image by UAV into simple as PlantVillage laboratory image. The edge node works on preprocessing and fore-background separation processes. The intended UAV captures images from the real-world environment and sends it to the edge node for being processed. In a future manuscript, we think the complete workflow scenario will be illustrated by real pictures across different phases of the classification process. Most drone operators are familiar with agricultural drone imaging. This study used drones in its data acquisition process. To begin, a local drone operator mapped the entire field.

2.2 Tomato leaf image preprocessing stage

Data augmentation and balancing are parts of the preprocessing phase. When training very little data, data augmentation will increase the dataset's size to several times that of the original, which helps in preventing overfitting. This method aids in creating more straightforward and more stable models that are generally applicable. The training dataset was augmented by replicating the available data with data rotation of ±30, the sheer range of 0.1, width shift range of 0.2, height shift range of 0.2, and zooms range of 0.3 to the horizontal. These parameters increased the training dataset size, improving performance and regularization, avoiding the overfitting problem. In data balancing, the number of images in each category was different, which means that the dataset was imbalanced, affecting the trained network's overall performance. To solve this problem, oversampling or under sampling techniques are used to obtain a balanced dataset. The class weight function, which applies an oversampling approach, was used in this work. Afterwards, the oversampling method was applied to the training set, making the dataset balanced at each category.

2.3 Features extraction stage

Applying the CapsNet architecture to the enrolled dataset provides the expected results because the original CapsNet used had fewer details and differences. In the proposed architecture, CapsNet's performance was improved by adding a layer to extract more features. This layer contains the first conv_2d layer of kernel size 3 with 64 filters, the second conv_2d layer of kernel size 3 with 168 filters, and the last conv_2d layer of kernel size 3 with 256 filters. Figure 2 shows the layers. The rectified linear unit (ReLU) was used as the nonlinear activation function after each layer, and the max-pooling layer was added after each convolution layer.

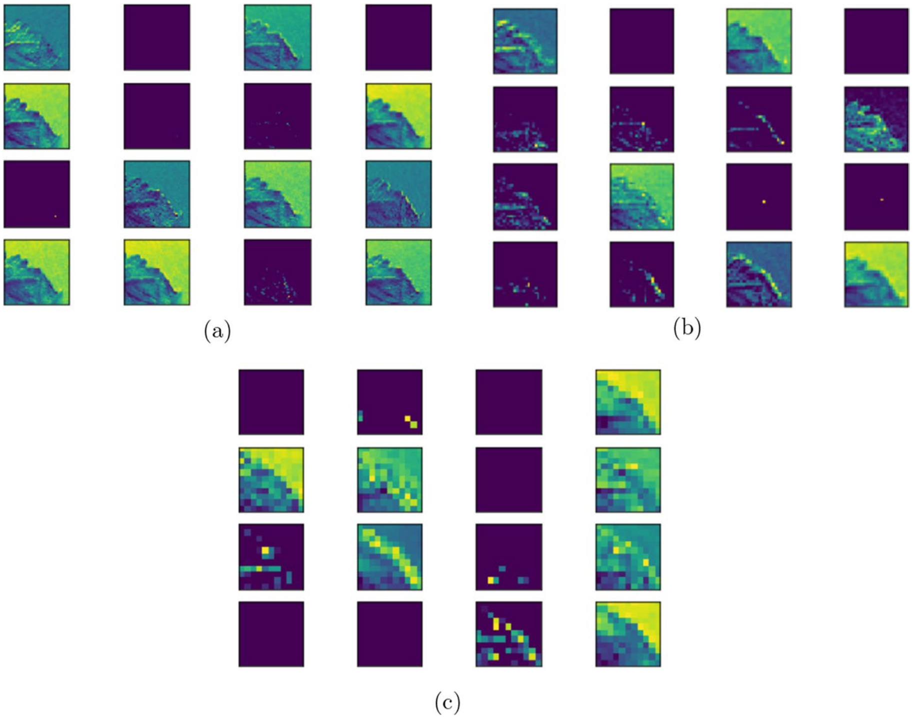


Fig. 2 **a** Feature extraction after the first layer, **b** feature extraction after the second layer, **c** and feature extraction after the third layer

2.4 Pruning feature stage

This stage comes after feature extraction to reduce the number of parameters that do not affect the system's performance. A fully connected layer with 1024 neurons and another fully connected layer with 512 neurons were used to get less weighted parameters. The output was reshaped to generate primary capsule output vectors, which comprise 32-layer capsules with 10 D. The output vectors were squashed using the squash-ing feature because the primary capsule layer was fully connected to the tomato capsule layer. A small epsilon value was applied to the squash function to avoid the vanishing gradient problem during training. Then, the output of the squash function was fed to the tomato capsule layer.

2.5 Capsule network (CapsNet) stage

The classification method for the resulting feature vectors from the pruning feature stage was conducted in this stage. The features from the first stage were remembered and reused in the second process. The second target task began as soon as the features were extracted. CapsNet was used and built on it because it is thought that the features extracted from a CNN trained dataset on a specific dataset could be helpful for another dataset with a similar issue. The CapsNet weights were updated and pretrained to produce vectors of the output feature extracted from this network that was transferred to the subsequent layers, called the primary capsule (PC) and dense capsule layers.

The PC layer comprises a convolutional layer taken after by a reshaping layer that reconfigures the excretory tensor into the capsule chain. In this layer, a dynamic routing algorithm was implemented.

The output of the capsules was passed through an activation function called "squashing," which was then passed on to the dense capsule layer to obtain a vector of the output feature of 16 dimensions of each class. With the L2 criterion, the output class was chosen as the target class. The margin loss Lc was the cost function or optimized error loss function in the capsules for each class C and vector class vc. A multiclass classification allowance was calculated for binary cross-entropy. Equation (5) was used to calculate the weights to compute the cost function:

$$\mathrm{L_{Co}} = \mathrm{T_{Co}} \max\left(0,\ \mathrm{mr}^{+} - ||\mathrm{V_{Co}}||\right)^{2} + \lambda(1 - \mathrm{T_{Co}}) \max\left(0, ||\mathrm{V_{Co}}|| - \mathrm{mr}^{-}\right)^{2} \tag{5}$$

where Tc = 1 corresponds to a class c image; otherwise, Tc = 0. During the experiments, m+ = 0.9 and m− = 0.1 were used. The "λ" gives a value = 0.5 and helps in stopping the initial learning values from reducing the classes' overall output vectors. The total loss for each class is equal to the sum of all losses as introduced by Mamidibathula et al. [61].

2.6 Adam optimizer

The expected value of a random variable to the power of n is called the N-th moment of that variable. It can be expressed as follows in Eqs. (6–15):

$$m_n = E\left[R^n\right] \tag{6}$$

where m is the moment, and R is the random variable.

The method that computes adaptive learning rates for every attribute is Adaptive Moment Estimation (Adam). Adam calculates exponentially moving averages based on the gradient of a current minibatch to estimate the moments. It preserves an exponential decay average of past squared gradients v_t in addition to an exponentially decaying average of the past gradient m_t. The mean is the first moment, and the uncentered variance is the second.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{7}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{8}$$

where m_t and v_t Are the estimates of the gradients' first moment (the mean) and the second moment (the uncentered variance). The default values for β_1 and β_2 are 0.9 and 0.999, respectively.

Adam enabled DL practitioners to significantly improve the optimization of existing algorithms over regular and stochastic gradient descent.

$$m_t = \varphi_1 m_{t-1} + (1 - \varphi_1) G_t \tag{9}$$

$$V_t = \varphi_2 V_{t-1} + (1 - \varphi_2) G_t^2 \tag{10}$$

where m and v are moving averages, G is the gradient on the current minibatch, and φ is the new algorithm hyperparameters introduced. The properties form and v are needed because they are estimates of the first and second moments:

$$E[m_t] = E[G_t] \tag{11}$$

$$E[V_t] = E\left[G_t^2\right] \tag{12}$$

The adjustment for these biases is obtained by computing bias-corrected estimates for the first and second moments as follows:

$$\hat{m}_t = \frac{m_t}{1 - \varphi_1^t} \tag{13}$$

$$\hat{V}_t = \frac{V_t}{1 - \varphi_2^t} \tag{14}$$

These bias-corrected estimates are then used to update the parameters, similar to Adadelta and RMSprop, resulting in the Adam update rule for updating weight:

$$w_t = w_{t-1} - \beta \frac{\hat{m}_t}{\sqrt{\hat{V}_t + \in}} \tag{15}$$

where w is the model weights and is the step size.

3 Experimental results: discussion and analysis

3.1 *Dataset description*

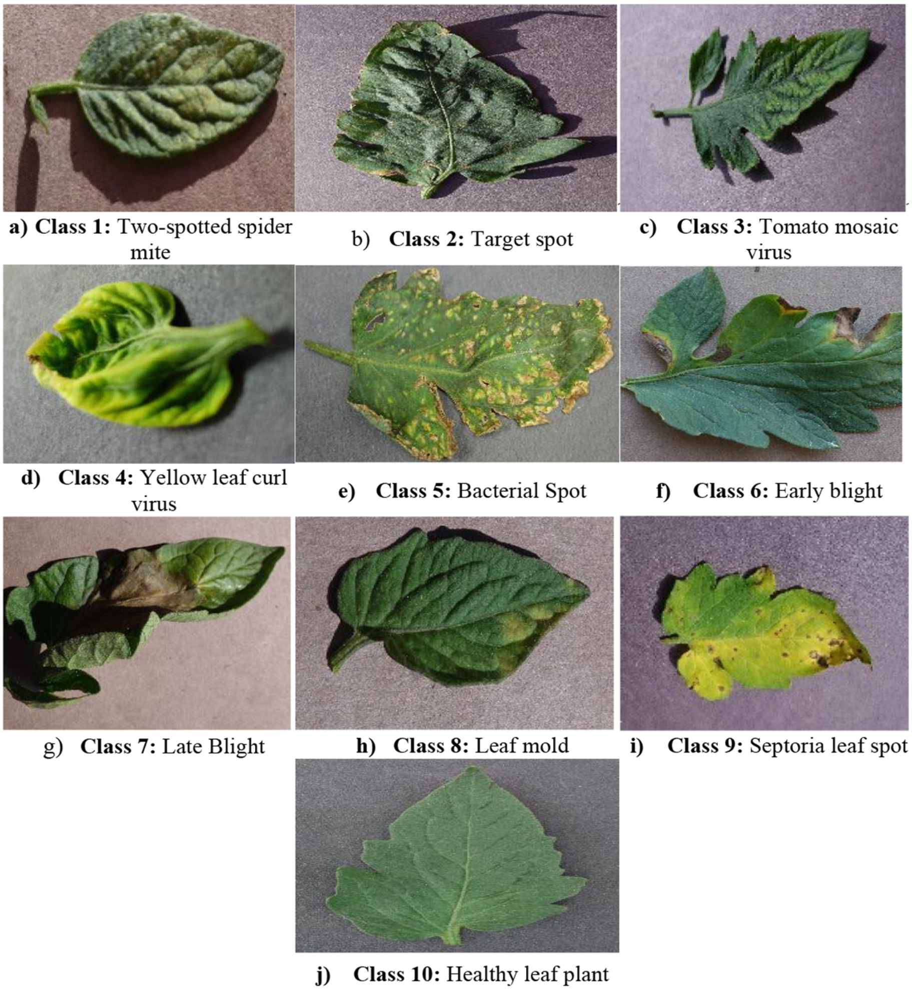
The plant disease images of tomatoes were used as the dataset for this work. It consists of 70,834 images of diseases and healthy tomato leaves' plants, 58,122 plant disease leaves for training and 12,712 images for testing. The images represent the tomatoes' leaf diseases in the leaves and the healthy tomato leaf. However, the dataset is characterized by distinguishing images taken at different angles and backgrounds. The dataset contains 10 categories: two-spotted spider mite, target spot, tomato mosaic virus, yellow leaf curl virus, bacterial Spot, early Blight, late Blight, leaf mold, septoria leaf spot, and healthy leaves. Table 1 shows the 10 class labels with the trained and tested numbers. Figure 3 shows the samples of 10 tomato leaf images offering the most common leaf diseases. The experimental results were obtained using tensor flow and Keras with GPU Google Colab environment. The two main experiments for this work are demonstrated as follows:

3.2 Experiment I: architecture without the capsule layer

In this experimental scenario, the proposed network architecture consists of three stages of convolution layers with "ReLU" activation function and max pooling, followed by two fully connected layers and finally a classification layer with "Soft-max" activation function.

Table 1 The 10 class labels for tomato leaf image diseases and the trained and tested samples

Class number	Class	Number of training samples	Number of testing samples
1	Two-spotted spider mite	5364	1340
2	Target spot	4496	1120
3	Tomato mosaic virus	1196	296
4	Yellow leaf curl virus	17,144	2484
5	Bacterial spot	6808	1700
6	Early blight	3200	800
7	Late blight	6109	1524
8	Leaf mold	3048	760
9	Septoria leaf spot	5668	1416
10	Healthy	5089	1272
Total		58,122	12,712



a) Class 1: Two-spotted spider mite

b) **Class 2:** Target spot

c) Class 3: Tomato mosaic virus

d) Class 4: Yellow leaf curl virus

e) Class 5: Bacterial Spot

f) Class 6: Early blight

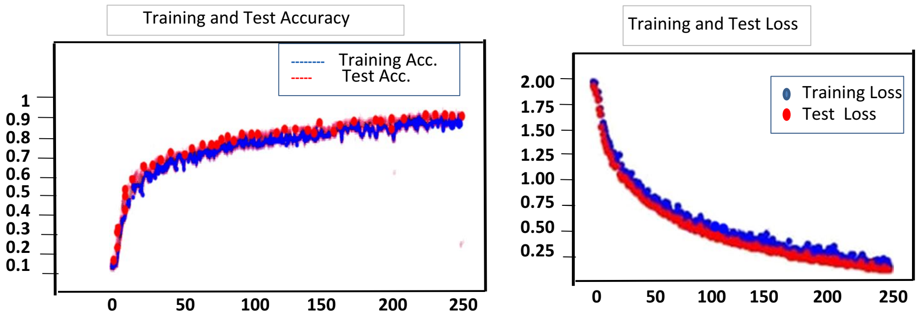
g) **Class 7:** Late Blight

h) Class 8: Leaf mold

i) Class 9: Septoria leaf spot

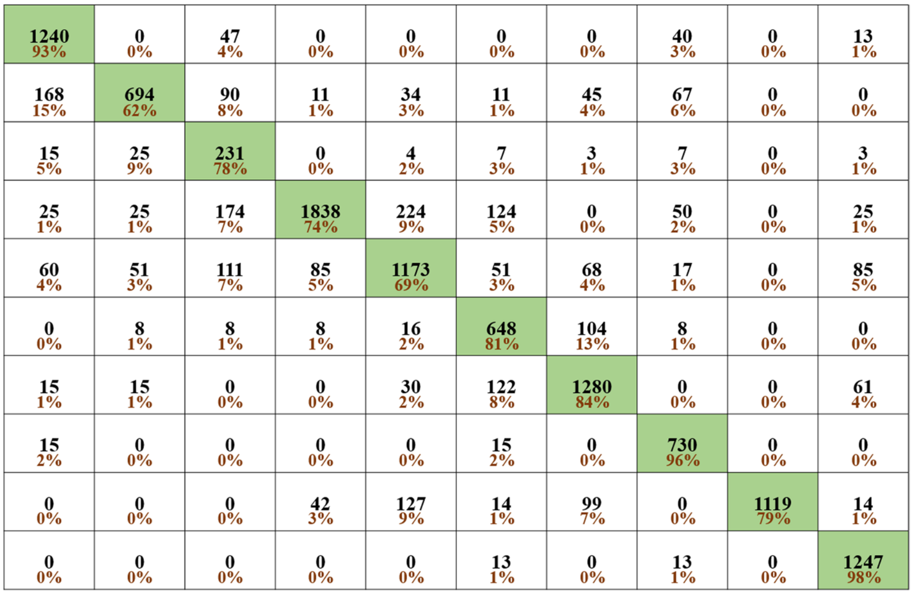
j) Class 10: Healthy leaf plant

Fig. 3 Samples of 10 classes of the tomato diseases image dataset



(a) (b)

Fig. 4 **a** Training and testing accuracy for Experiment-I without CapsNet, and **b** training loss and testing loss for Experiment-I without CapsNet



1240 93%	0 0%	47 4%	0 0%	0 0%	0 0%	0 0%	40 3%	0 0%	13 1%
168 15%	694 62%	90 8%	11 1%	34 3%	11 1%	45 4%	67 6%	0 0%	0 0%
15 5%	25 9%	231 78%	0 0%	4 2%	7 3%	3 1%	7 3%	0 0%	3 1%
25 1%	25 1%	174 7%	1838 74%	224 9%	124 5%	0 0%	50 2%	0 0%	25 1%
60 4%	51 3%	111 7%	85 5%	1173 69%	51 3%	68 4%	17 1%	0 0%	85 5%
0 0%	8 1%	8 1%	8 1%	16 2%	648 81%	104 13%	8 1%	0 0%	0 0%
15 1%	15 1%	0 0%	0 0%	30 2%	122 8%	1280 84%	0 0%	0 0%	61 4%
15 2%	0 0%	0 0%	0 0%	0 0%	15 2%	0 0%	730 96%	0 0%	0 0%
0 0%	0 0%	0 0%	42 3%	127 9%	14 1%	99 7%	0 0%	1119 79%	14 1%
0 0%	0 0%	0 0%	0 0%	0 0%	13 1%	0 0%	13 1%	0 0%	1247 98%

Fig. 5 Experiment-I's confusion matrix of the 10 class labels representing tomato leaf diseases

Figure 4 shows training and testing accuracy, and Fig. 5 shows the confusion matrix indicating the testing classification accuracy of 92.87%. As shown in Fig. 4, using 250 epochs, both accuracy and loss function were determined. It also shows the increase in the accuracy to achieve 92.87% and the decrease in the loss of 0.255. It can be observed from Fig. 5 that 92% of class 1 means that the two-spotted spider mite disease was correctly classified, 62% of class 2 means that target spot disease was correctly classified, and so on. In this experiment, the healthy leaves of the tomato plant were 98% correctly classified.

3.3 Experiment II: simple CapsNet architecture with three convolutional layers

To improve the accuracy of the recognition process for tomato leaf disease, this experiment was initially conducted using a simple CapsNet architecture with three layers as proposed by Sabour et al. [40]. Because spots on the leaf surface identify tomato diseases and the location of this point varies from disease to disease, the capsule examination was expected to give the best results because it is concerned with the spatial relationship features of the image (as mentioned earlier).

Following up on the results, it was discovered that the results were unstable: may be very high, may reach 100% or very weak, and may reach 40% or less. Thus, the network architecture was modified to obtain stable performance with high accuracy. By applying the proposed architecture based on CapsNet (illustrated in Fig. 1), the network architecture consists of three stages of convolution layers with "ReLU" activation function and max pooling, followed by two fully connected layers and finally CapsNet, CapsNet consisted of primary and tomato capsule layers.

The execution of this model was accomplished using an Adam optimizer with a 0.00001 learning rate, the dimension of the capsules in the PC layer was 10, the number of capsules in tomato capsule layers was 16, the number of routine iterations was 4, and the overall epochs of training was 250. Using CapsNet helped determine the location

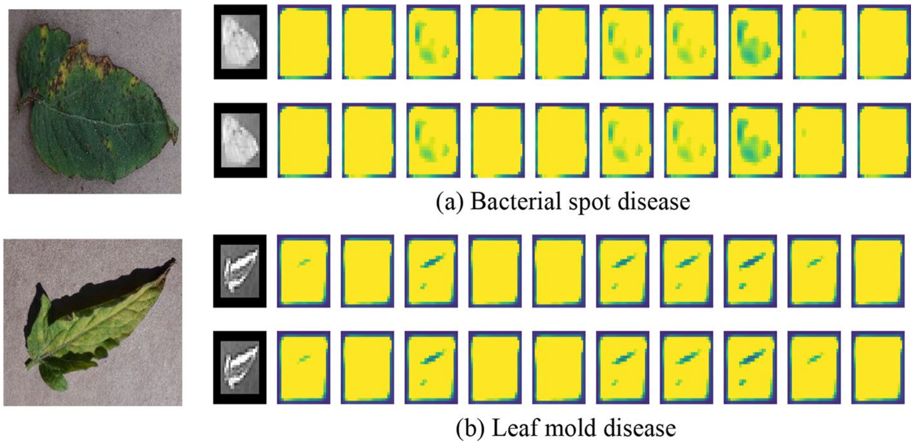


Fig. 6 Visual representation of the plant diseases (**a**) Bacterial spot disease, and (**b**) Leaf mold disease

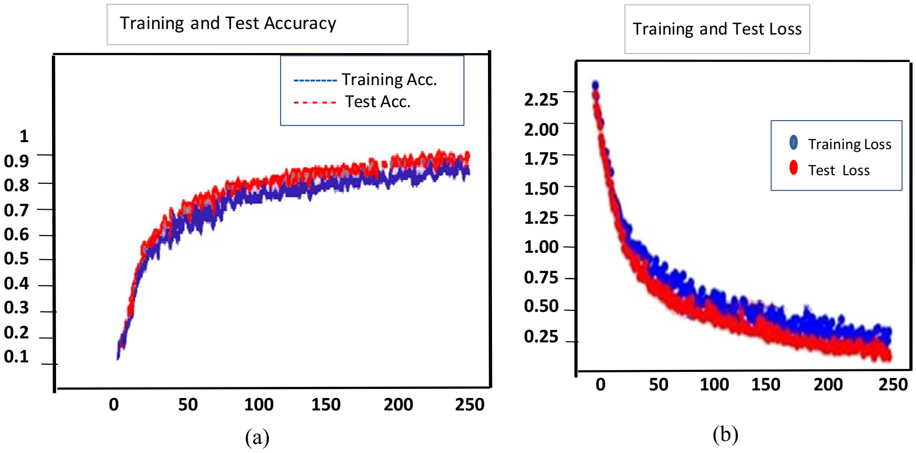
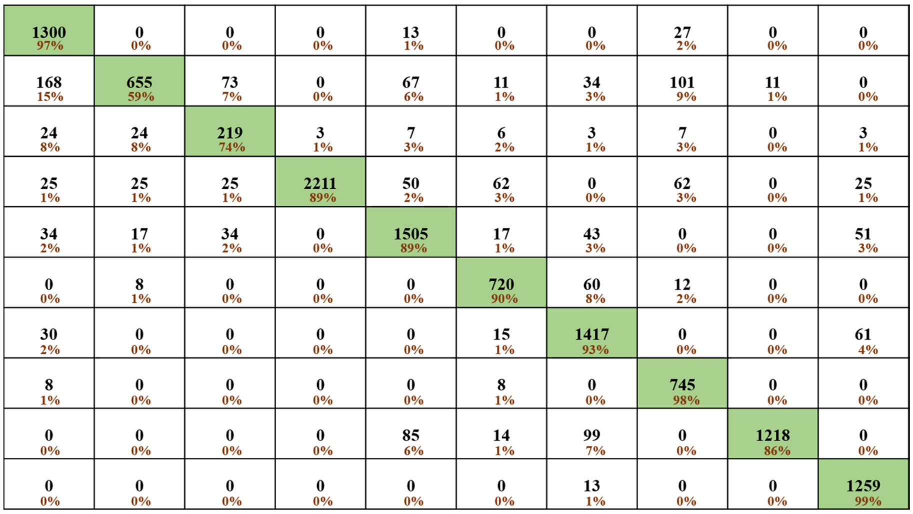


Fig. 7 **a** Training and testing accuracy for Experiment II with CapsNet, and **b** training and testing loss for Experiment II with CapsNet



1300 97%	0 0%	0 0%	0 0%	13 1%	0 0%	0 0%	27 2%	0 0%	0 0%
168 15%	655 59%	73 7%	0 0%	67 6%	11 1%	34 3%	101 9%	11 1%	0 0%
24 8%	24 8%	219 74%	3 1%	7 3%	6 2%	3 1%	7 3%	0 0%	3 1%
25 1%	25 1%	25 1%	2211 89%	50 2%	62 3%	0 0%	62 3%	0 0%	25 1%
34 2%	17 1%	34 2%	0 0%	1505 89%	17 1%	43 3%	0 0%	0 0%	51 3%
0 0%	8 1%	0 0%	0 0%	0 0%	720 90%	60 8%	12 2%	0 0%	0 0%
30 2%	0 0%	0 0%	0 0%	0 0%	15 1%	1417 93%	0 0%	0 0%	61 4%
8 1%	0 0%	0 0%	0 0%	0 0%	8 1%	0 0%	745 98%	0 0%	0 0%
0 0%	0 0%	0 0%	0 0%	85 6%	14 1%	99 7%	0 0%	1218 86%	0 0%
0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	13 1%	0 0%	0 0%	1259 99%

Fig. 8 Experiment-II's confusion matrix of the 10 class labels representing tomato leaf diseases

of the infection within the plant leaf, where the type of disease varies according to the location of the infection. Figure 6 shows the visual representation of the capsule layer for bacterial spots and leaf mold diseases.

The experimental results indicated that the training and testing accuracies were more stable and higher than previous results. As shown in Fig. 7, the proposed architecture's accuracy was 96.39%, with a minimum loss function of 0.221. Moreover, the confusion matrix for 10 classes was determined, and each class contained 10 leaf images of tomato diseases. As investigated in the confusion matrix, a significant enhancement was achieved compared with a traditional CNN demonstrated in Experiment I. For example, 99% observed for healthy tomato leaf images were correctly classified, as shown in Fig. 8.

3.4 Performance evaluation and discussions

Five evaluation indexes were used in the classification problem: accuracy, precision, recall, F1-score, and confusion matrix. These indexes were used to evaluate the proposed approach's predictive ability. Accuracy is defined as the ratio of correct predictions to total predictions, expressed as a percentage. It was calculated using Eq. (16). Precision is a factor that calculates a model's ability to predict values for a specific category correctly, and it was measured using Eq. (17). Recall measured the fraction of accurately categorized positive patterns. This was determined using Eq. (18). The weighted average of precision and recall was the F1-score. The macro standard was used to measure the accuracy, recall, and F1-score's overall results. By mapping expected outputs over actual outputs, the confusion matrix is a table that is commonly used to explain the performance of a classification model on a test set for which the correct values are known [58, 59].

Table 2 illustrates the five evaluation measures for the proposed model with CapsNet. From Fig. 1, it was ensured that when compared to the traditional CNN model (without CapsNet), the proposed model gave a better performance without CapsNet. Additionally, as shown in Table 3, how much the proposed model outperforms previous works were compared. Although Abbas et al., [48] achieved higher accuracies 97.11%, than the proposed method. They utilized CGAN algorithm with DensNet121 by which the

Table 2 The performance measurement results for the traditional CNN architecture and the proposed model with CapsNet

Class number	Ordinary architecture without CapsNet				Proposed model with CapsNet			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
1	92.00	82.60	92.30	87.18	97.00	82.40	96.60	88.94
2	62.00	68.00	61.60	64.64	59.00	73.10	58.80	65.17
3	78.00	80.90	77.60	79.22	74.00	91.70	74.40	82.15
4	74.00	82.70	74.50	78.39	69.00	96.70	69.30	80.74
5	69.00	82.00	69.00	74.94	88.00	81.70	88.10	84.78
6	81.00	78.00	81.10	79.52	90.00	84.70	90.40	87.46
7	84.00	75.00	83.90	79.20	83.00	82.90	82.90	82.90
8	96.00	95.80	95.50	95.65	98.00	95.60	97.80	96.69
9	79.00	95.90	79.10	86.69	86.00	95.50	85.80	90.39
10	98.00	89.90	98.10	93.82	99.00	91.80	98.80	95.17

Table 3 A comparative study of the proposed architecture compared with previous work for tomato disease recognition

Researchers	Methodology	Dataset specification	Class labels	Accuracy
Mokhtar et al., 2015 [45]	Support Vector Machine (SVM)	The 200 infected tomato leaf images dataset includes two different yellow leaf curl diseases	Two Yellow Leaf Diseases	90.00%
Brahimi et al., 2017 [47]	Using CNN compared with shallow models and hand-crafted features	The dataset contains 14,828 images for nine diseases	4032 Yellow Leaf Curl Virus 1. 325 Mosaic Virus 2. 1356 Target Spot 3. 1628 Spider Mites 4. 904 Leaf Mold 5. 1723 Septoria Spot 6. 1781 Late Blight 7. 952 Early Blight 2127 Bacterial Spot	94.53%, and 95.46%
Sardogan et al., 2018) [44]	CNN model and Learning Vector Quantization (LVQ) algorithm	The dataset with 400 leaf images for training and 100 images for testing the following diseases	Healthy Bacterial Spot Late Blight Septoria Spot, Yellow Curved	86.00%
Zhang et al., 2018 [60]	CNN and the applied transfer learning algorithms for Resnet, Alexnet, and GoogleNet	The dataset contains 5550 images for Health and other eight diseases included	Early Blight, Yellow Leaf Curl Disease, Yellow Leaf Curl Virus, Leaf Spot, Leaf Mold disease, Mosaic Virus, Late Blight, Two-spotted Spider Mite	96.51%
Foysal et al., 2020 [46]	CNN	The dataset has 600 input images, 100 for each class, and six class leaf diseases	Healthy Bacterial Spot Late Blight Septoria Spot, Yellow Curved Spider Mites	76.00%

Table 3 (continued)

Researchers	Methodology	Dataset specification	Class labels	Accuracy
Abbas et al., 2021 [48]	Conditional Generative Adversarial Network (C-GAN) with DenseNet121 model	PlantVillage dataset contains ten categories of diseases with 16,012 images	Tomato Yellow Leaf Curl Virus Tomato Bacterial Spot Tomato Late Blight Tomato Septoria leaf spot Tomato Two Spotted Spider Mite Tomato Target Spot Tomato Early Blight Tomato Leaf Mold Tomato Mosaic Virus Tomato healthy	DenseNet, C-GAN 97.11%
Atila et al., 2021 [49]	EfficientNet	PlantVillage dataset contains 39 plant leaf diseases and 10 class labels of tomato leaf images The number of tested images of tomatoes was 500 leaf images	Tomato Bacterial Spot Tomato Early blight Tomato Late blight Tomato Leaf Mold Tomato Septoria leaf spot Tomato Spider mites Tomato Target Spot Tomato Yellow Leaf Curl Tomato mosaic virus Tomato healthy	Average accuracy 99.00%
Chowdhury et al., 2021 [50]	EfficientNet and modified U-net	PlantVillage dataset contains 16,485 images with ten class labels	Healthy Early Blight Septoria Leaf Spot Target Spot Leaf Mold Bacterial Spot Late Bright Mold Tomato Yellow Leaf Curl Virus Tomato Mosaic Virus	Accuracy 99.17%

Tan et al., 2021 [51]	KNN
	SVM
	RF
	AlexNet
	VGG16
	ResNet34
	EffeicientNet
	MobileNetV2
		Ten class labels of PlantVillage dataset with 1591 healthy and 5357 infected tomato images	Bacterial Spot
			Early Blight
			Late Blight
			Leaf mold
			Septoria leaf spot
			Two-spotted spider mite
			Target spot
			Tomato mosaic virus
			Tomato yellow leaf curl virus
			Health
				Accuracy
				KNN = 82.10%
				SVM = 91.00%
				RF = 82.70%
				AlexNet = 92.70%
				VGG16 = 98.90%
				ResNet34 = 99.70%
				EfficientNet = 98.90%
				MobileNetV2 = 91.20%

Proposed Architecture	Traditional CNN and Capsule Network-based Adam optimizer	The dataset contains 10 categories; Trained images • 58,122 Tested images • 12,712 The total number of images • 70,834	Two-spotted Spider Mite, Target Spot, Tomato Mosaic Virus, Yellow Leaf Curl Virus, Bacterial Spot, Early Blight, Late Blight, Leaf Mold, Septoria Leaf Spot Healthy Leaves	Traditional CNN = 92.87% Capsule network accuracy = 96.39%

dataset augmentation for synthetic images is performed on the PlantVillage dataset with 16,012 images. Furthermore, Atila, [49] achieved higher accuracy than the proposed model; they used the limited number of tested images representing the 39 classes 1950 images for all kinds of plants and only 500 tested tomato leaf images. To get more reliable findings, they need to expand their dataset by considering the plant diversity and the number of classes.

The proposed model used a dataset of 70,834 images (for training and testing) for 10 different tomato leaf diseases and did not use any pre-train models.

$$\text{Accuracy} = \frac{\text{Number of accurate predictions}}{\text{Total number of prediction}} \tag{16}$$

$$\text{Precision} = \frac{\text{particular category predicted correctly}}{\text{all category predictions}} \tag{17}$$

$$\text{Recall} = \frac{\text{Category was correctly predicted}}{\text{All real categories}} \tag{18}$$

4 Conclusion and future work

In this paper, we proposed an effective and robust architecture based on the optimized Capsule Neural Network (CapsNet) for the classification and recognition of different tomato leaf diseases. Our methodology focused on detecting common diseases that affect the surface of plant leaves using images captured by drones. To validate the performance of the proposed CapsNet approach, we utilized a large-scale dataset comprising 70,834 images and compared it with traditional Convolutional Neural Networks (CNNs). The proposed architecture successfully addressed the limitations associated with traditional CNNs, such as unstable results and slight performance decrease. By leveraging the advantages of CapsNet, our approach achieved an accuracy of 96.39% with a minimum loss rate of 0.221, outperforming the traditional CNN approach, which achieved an accuracy of 92.87%. This significant improvement in accuracy highlights the effectiveness of CapsNet in accurately identifying and classifying ten different tomato leaf diseases, including two-spotted spider mite, target spot, tomato mosaic virus, yellow leaf curl virus, bacterial spot, early blight, late blight, leaf mold, septoria leaf spot, and healthy leaves.

In future research, we plan to utilize unmanned aerial vehicles (UAVs) to collect plant leaf images, enabling more efficient and comprehensive monitoring of diseases. The collected images will undergo various preprocessing and transformation stages to handle the complexities introduced by the drone's background. Additionally, we recommend expanding the scope of this study to include different types of plants and investigate additional features that can effectively represent diseases in plant leaves. By addressing these challenges, we aim to contribute to the development of early disease management strategies and overcome one of the fundamental challenges in ensuring food quality measurements.

Abbreviations

CNN	Convolutional Neural Networks
CapsNet	Capsule Network
UAV	Unmanned aerial vehicles
KNN	K-nearest neighbor
SVM	Support vector machine
DL	Deep learning
LVQ	Learning vector quantization
EffecientNet	Efficient networks
ReLU	Rectified linear unit
Adam	Adaptive moment
RF	Random forest
CGAN	Conditional generative adversarial network
VGG	Visual geometry group version 16
ResNet	Residual network
MobileNetV2	Mobile NETWORKS

Acknowledgements
Lobna M. Abouelmagd, Mahmoud Y. Shams, Aboul Ella Hassanien—Scientific Research School of Egypt (SRSEG), http://www.egyptscience.net.

Author contributions
All authors are equally contributed.

Funding
Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Availability of data and materials
Data is available as referenced in the text.

Code availability
Available on request.

Declarations

Competing interests
Not applicable.

Received: 1 October 2022 Accepted: 28 December 2023
Published online: 08 January 2024

References

1. H. Wu, L. Fang, Q. Yu, J. Yuan, C. Yang, Plant leaf identification based on shape and convolutional features. Expert Syst. Appl. **219**, 119626 (2023)
2. T. Wiesner-Hanks, H. Wu, E. Stewart, C. DeChant, N. Kaczmar, H. Lipson, M.A. Gore, R.J. Nelson, Millimeter-level plant disease detection from aerial photographs via deep learning and crowdsourced data. Front. Plant Sci. **10**, 1550 (2019)
3. K. Neupane, F. Baysal-Gurel, Automatic identification and monitoring of plant diseases using unmanned aerial vehicles: a review. Remote Sens. **13**, 3841 (2021)
4. U. Shruthi, V. Nagaveni, C. S. Arvind, G. L. Sunil, *Tomato plant disease classification using deep learning architectures: a review*. Proceedings of second international conference on advances in computer engineering and communication systems: ICACECS 2021 (Springer, 2022), pp. 153–169.
5. V. harun, S. Parthiban, T. B. Marry, M. Sagayam, A. A. Elngar, *Future trends and challenges of UAV: conclusion*. https://doi.org/10.4018/978-1-7998-8763-8.ch010.
6. A. Hafeez, M. A. Husain, S. P. Singh, A. Chauhan, Mohd. T. Khan, N. Kumar, A. Chauhan, S. K. Soni, Implementation of drone technology for farm monitoring & pesticide spraying: A review. Inf. Process. Agric. **10**, 192 (2022).
7. S.D. Apostolidis, PCh. Kapoutsis, ACh. Kapoutsis, E.B. Kosmatopoulos, Cooperative multi-UAV coverage mission planning platform for remote sensing applications. Auton. Robots **46**, 373 (2022)
8. D. Gao, Q. Sun, B. Hu, S. Zhang, A Framework for agricultural pest and disease monitoring based on internet-of-things and unmanned aerial vehicles. Sensors **20**, 1487 (2020)
9. N. Kaur, S. Verma, N. Z. Jhanjhi, *De-noising diseased plant leaf image*. In: 2022 2nd international conference on computing and information technology (ICCIT) (IEEE, 2022), pp. 130–137.
10. M.Y. Shams, O.M. Elzeki, L.M. Abouelmagd, A.E. Hassanien, M. Abd Elfattah, H. Salem, HANA: a healthy artificial nutrition analysis model during COVID-19 pandemic. Comput. Biol. Med. **135**, 104606 (2021)
11. S.H. Lee, C.S. Chan, S.J. Mayo, P. Remagnino, How deep learning extracts and learns leaf features for plant classification. Pattern Recognit. **71**, 1 (2017)
12. G. Saleem, M. Akhtar, N. Ahmed, W.S. Qureshi, Automated analysis of visual leaf shape features for plant classification. Comput. Electron. Agric. **157**, 270 (2019)

13. A. Kaya, A.S. Keceli, C. Catal, H.Y. Yalic, H. Temucin, B. Tekinerdogan, Analysis of transfer learning for deep neural network based plant classification models. Comput. Electron. Agric. **158**, 20 (2019)
14. M.A. Chandra, S.S. Bedi, *Classification of plant based on leaf images, in advances in computational intelligence and communication technology* (Springer, 2021), pp.29–37
15. M. Keivani, J. Mazloum, E. Sedaghatfar, M.B. Tavakoli, Automated analysis of leaf shape, texture, and color features for plant classification. Trait. Signal **37**, 17–28 (2020)
16. M. K. Choudhary, S. Hiranwal, *Feature selection algorithms for plant leaf classification: a survey*. Proceedings of international conference on communication and computational technologies, edited by S. D. Purohit, D. Singh Jat, R. C. Poonia, S. Kumar, and S. Hiranwal (Springer, Singapore, 2021), pp. 657–669.
17. A. Afifi, A. Alhumam, A. Abdelwahab, Convolutional neural network for automatic identification of plant diseases with limited data. Plants **10**, 28 (2020)
18. N.E.M. Khalifa, M.H.N. Taha, L.M. El-Maged, A.E. Hassanien, Artificial intelligence in potato leaf disease classification: a deep learning approach, in *Machine learning and big data analytics paradigms: analysis, applications and challenges*. (Springer, New York, 2021), pp.63–79
19. V. Singh, A.K. Misra, Detection of plant leaf diseases using image segmentation and soft computing techniques. Inf. Process. Agric. **4**, 41 (2017)
20. M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, D. Stefanovic, Solving current limitations of deep learning based approaches for plant disease detection. Symmetry **11**, 939 (2019)
21. A. Devaraj, K. Rathan, S. Jaahnavi, K. Indira, *Identification of plant disease using image processing technique*. 2019 International conference on communication and signal processing (ICCSP) (IEEE, 2019), pp. 0749–0753.
22. H. Salem, G. Attiya, N. El-Fishawy, *Gene expression profiles based human cancer diseases classification*. 2015 11th international computer engineering conference (ICENCO) (2015), pp. 181–187.
23. H. Waghmare, R. Kokare, Y. Dandawate, *Detection and classification of diseases of grape plant using opposite colour local binary pattern feature and machine learning for automated decision support system*. 2016 3rd international conference on signal processing and integrated networks (SPIN) (2016), pp. 513–518.
24. S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M.P. Reyes, M.-L. Shyu, S.-C. Chen, S.S. Iyengar, A survey on deep learning: algorithms, techniques, and applications. ACM Comput. Surv. **51**, 92 (2018)
25. Z.-Q. Zhao, P. Zheng, S. Xu, X. Wu, Object detection with deep learning: a review, arXiv:1807.05511.
26. S. Albawi, T. A. Mohammed, S. Al-Zawi, *Understanding of a convolutional neural network*. 2017 International Conference on Engineering and Technology (ICET) (Ieee, 2017), pp. 1–6.
27. O.M. Elzeki, M. Shams, S. Sarhan, M. Abd Elfattah, A.E. Hassanien, COVID-19: a new deep learning computer-aided model for classification. PeerJ Comput. Sci. **7**, e358 (2021)
28. M. Y. Shams, O. M. Elzeki, M. Abd Elfattah, T. Medhat, A. E. Hassanien, *Why are generative adversarial networks vital for deep neural networks? A case study on COVID-19 chest X-ray images*. Big data analytics and artificial intelligence against COVID-19: innovation vision and approach (Springer, 2020), pp. 147–162.
29. G. Zoumpourlis, A. Doumanoglou, N. Vretos, P. Daras, *Non-linear convolution filters for CNN-based learning* (2017), pp. 4761–4769.
30. T. S. Cohen, M. Geiger, J. Köhler, M. Welling, *Spherical Cnns*, ArXiv Prepr. arXiv:1801.10130 (2018).
31. K. Han, H. Wen, J. Shi, K.-H. Lu, Y. Zhang, D. Fu, Z. Liu, Variational autoencoder: an unsupervised model for encoding and decoding FMRI activity in visual cortex. Neuroimage **198**, 125 (2019)
32. V. Andrearczyk, J. Fageot, V. Oreiller, X. Montet, A. Depeursinge, *Exploring local rotation invariance in 3D CNNs with steerable filters*. International conference on medical imaging with deep learning (PMLR, 2019), pp. 15–26.
33. M.E. ElAraby, O.M. Elzeki, M.Y. Shams, A. Mahmoud, H. Salem, A novel gray-scale spatial exploitation learning net for COVID-19 by crawling internet resources. Biomed. Signal Process. Control **73**, 103441 (2022)
34. E.C. Too, L. Yujian, S. Njuki, L. Yingchun, A Comparative study of fine-tuning deep learning models for plant disease identification. Comput. Electron. Agric. **161**, 272 (2019)
35. H. Salem, M.Y. Shams, O.M. Elzeki, M. Elfattah, J.F. Al-Amri, S. Elnazer, Fine-tuning fuzzy KNN classifier based on uncertainty membership for the medical diagnosis of diabetes. Appl. Sci. **12**, 950 (2022)
36. V.A. Sindagi, V.M. Patel, A survey of recent advances in CNN-based single image crowd counting and density estimation. Pattern Recognit. Lett. **107**, 3 (2018)
37. A. D. Kumar, *Novel deep learning model for traffic sign detection using capsule networks*, arXiv:1805.04424.
38. M. Kennelly, J. O'Mara, C. Rivard, G.L. Miller, D. Smith, Introduction to abiotic disorders in plants. Plant Health Instr. **10**, 10 (2012)
39. G. Altan, Performance evaluation of capsule networks for classification of plant leaf diseases. Int. J. Appl. Math. Electron. Comput. **8**, 3 (2020)
40. S. Sabour, N. Frosst, G.E. Hinton, *Dynamic routing between capsules, in advances in neural information processing systems*, vol. 30 (Curran Associates Inc, 2017)
41. D. P. Kingma, J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980.
42. I. Pagán, M. del Carmen Córdoba-Sellés, L. Martínez-Priego, A. Fraile, J.M. Malpica, C. Jordá, F. García-Arenal, Genetic structure of the population of pepino mosaic virus infecting tomato crops in Spain. Phytopathology **96**, 274 (2006)
43. R. Sujatha, J.M. Chatterjee, N.Z. Jhanjhi, S.N. Brohi, Performance of deep learning vs machine learning in plant leaf disease detection. Microprocess. Microsyst. **80**, 103615 (2021)
44. M. Sardogan, A. Tuncer, Y. Ozen, *Plant leaf disease detection and classification based on CNN with LVQ algorithm*. 2018 3rd international conference on computer science and engineering (UBMK) (2018), pp. 382–385.
45. U. Mokhtar, M.A.S. Ali, A.E. Hassanien, H. Hefny, Identifying two of tomatoes leaf viruses using support vector machine, in *Information systems design and intelligent applications*. ed. by J.K. Mandal, S.C. Satapathy, M. Kumar Sanyal, P.P. Sarkar, A. Mukhopadhyay (Springer India, New Delhi, 2015), pp.771–782
46. F. A. Foysal, M. Shakirul Islam, S. Abujar, S. Akhter Hossain, *A novel approach for tomato diseases classification based on deep convolutional neural networks*. Proceedings of International Joint Conference on Computational Intelligence (Springer, 2020), pp. 583–591.

47. M. Brahimi, K. Boukhalfa, A. Moussaoui, Deep learning for tomato diseases: classification and symptoms visualization. Appl. Artif. Intell. **31**, 299 (2017)
48. A. Abbas, S. Jain, M. Gour, S. Vankudothu, Tomato plant disease detection using transfer learning with C-GAN synthetic images. Comput. Electron. Agric. **187**, 106279 (2021)
49. Ü. Atila, *Uçar M Akyol K Uçar E*, Plant Leaf Dis. Classif. Using Effic. Deep Learn. Model Ecol Inf. **61**, 10.1016 (2021).
50. M.E. Chowdhury, T. Rahman, A. Khandakar, M.A. Ayari, A.U. Khan, M.S. Khan, N. Al-Emadi, M.B.I. Reaz, M.T. Islam, S.H.M. Ali, Automatic and reliable leaf disease detection using deep learning techniques. AgriEngineering **3**, 294 (2021)
51. L. Tan, J. Lu, H. Jiang, Tomato leaf diseases classification based on leaf images: a comparison between classical machine learning and deep learning methods. AgriEngineering **3**, 3 (2021)
52. B. Li, M.Q.-H. Meng, Texture analysis for ulcer detection in capsule endoscopy images. Image Vis. Comput. **27**, 1336 (2009)
53. E. Xi, S. Bing, Y. Jin, *Capsule Network Performance on Complex Data*, arXiv:1712.03480.
54. G. Sun, S. Ding, T. Sun, C. Zhang, W. Du, A novel dense capsule network based on dense capsule layers. Appl. Intell. **52**, 3066 (2022)
55. P. R. Ananya, V. Pachisia, S. Ushasukhanya, *Optimization of CNN in capsule networks for Alzheimer's disease prediction using CT images*. Proceedings of International conference on deep learning, computing and intelligence, edited by G. Manogaran, A. Shanthini, and G. Vadivu (Springer Nature, Singapore, 2022), pp. 551–560.
56. H. Sharma, A.S. Jalal, A Survey of methods, datasets and evaluation metrics for visual question answering. Image Vis. Comput. **116**, 104327 (2021)
57. L.M. AbouEl-Magd, A. Darwish, V. Snasel, A.E. Hassanien, A pre-trained convolutional neural network with optimized capsule networks for chest X-rays COVID-19 diagnosis. Clust. Comput. **26**, 1389–1403 (2022)
58. S. Sarhan, A.A. Nasr, M.Y. Shams, Multipose face recognition-based combined adaptive deep learning vector quantization. Comput. Intell. Neurosci. (2020). https://doi.org/10.1155/2020/8821868
59. H. Salem, G. Attiya, N. El-Fishawy, *Intelligent decision support system for breast cancer diagnosis by gene expression profiles*. 2016 33rd National Radio Science Conference (NRSC) (2016), pp. 421–430.
60. K. Zhang, Wu. Qiufeng, A. Liu, X. Meng, Can deep learning identify tomato leaf disease? Adv. Multimed. **2018**, 1 (2018)
61. B. Mamidibathula, S. Amirneni, S.S. Sistla, and N. Patnam, Texture classification using capsule networks, in Pattern Recognition and Image Analysis, edited by A. Morales, J. Fierrez, J.S. Sánchez, and B. Ribeiro, Lecture Notes in Computer Science (Springer International Publishing, Cham, 2019), pp. 589–599.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.