ORIGINAL ARTICLE

Tomato fruit disease detection based on improved single shot detection algorithm

Benedicta Nana Esi Nyarko*, Wu Bin, Zhou Jinzhi, Justice Odoom

School of Information Engineering, Southwest University of Science and Technology, Mianyang, Sichuan, China

Vol. 63, No. 4: 405–417, 2023

DOI: 10.24425/jppr.2023.146877

Received: May 17, 2023
Accepted: July 06, 2023
Online publication: October 02, 2023

*Corresponding address:
benedictanyarko41@gmail.com

Responsible Editor:
Rafał Kukawka

Abstract

The tomato crop is more susceptible to disease than any other vegetable, and it can be infected with over 200 diseases caused by different pathogens worldwide. Tomato plant diseases have become a challenge to food security globally. Currently, diagnosing and preventing tomato plant diseases is a challenge due to the lack of essential methods or tools. The traditional techniques of detecting plant disease are arduous and error-prone. Utilizing precise or automatic detection methods in spotting early plant disease can improve the quality of food production and reduce adverse effects. Deep learning has significantly increased the recognition accuracy of image classification and object detection systems in recent years. In this study, a 15-layer convolutional neural network is proposed as the backbone for single shot detector (SSD) to improve the detection of healthy, and three classes of tomato fruit diseases. The proposed model performance is compared with ResNet-50, AlexNet, VGG 16, and VGG19 as the backbone for Single shot detector. The findings of the experiment showed that the proposed CNN-SDD achieved 98.87% higher detection accuracy, which outperformed state-of-the-art models.

Keywords: convolutional neural network, deep learning, feature extraction, model backbone, plant disease detection, single shot detector algorithm

Introduction

Modern technologies can produce enough food to meet the requirements of over 7 billion people (Hemathilake and Gunathilake 2022). Yet, several issues, such as decreasing pollinators, climate change, and plant disease continually pose a threat to food stability (Hofman-Bergholm 2023). In addition to endangering the safety of the world's food supply, plant diseases can have tragic effects on smallholder farmers whose livelihoods depend on robust crops. Several initiatives have been created to stop crop loss from diseases (Khan *et al.* 2023). The integrated pest management approach has largely substituted conventional techniques for applying insecticides in the past 10 years (Golan *et al.* 2023). The environment, human health, and plant health are all harmed by the extensive use of these chemical techniques. Additionally, these techniques raise production costs. Plant disease management is effective when diseases are detected at an early stage.

Lycopersicon esculentum (Mohan *et al.* 2023) (Tomato) previously referred to as *Solanum Lycopersicon* L (Leite and Fialho 2018) belongs to the family *Solanaceae* (Knapp and Peralta 2016) and emanates from the Andes region of South America. Current global research indicates that approximately 182 million tons of tomato crops are cultivated from 5 million hectares of land (Caruso 2022, Bhujel *et al.* 2022). With 0.2 million hectares, it is the leading vegetable with the largest yield in southern

Europe. Turkey, Egypt, and Italy are the leading producers. This is a blooming species, most frequently a sprawling and nightshade plant that is usually grown for its edible fruit (Liu *et al.* 2022).

Phonological changes in the tomato plant can include aberrant growth, pigmentation, spots, deformities, wilting, desiccation, and necrosis on the leaves (Humbal and Pathak 2023, Kremneva *et al.* 2023; Nkongho *et al.* 2023; Sreedevi and Manike 2023). Tomato disease control is best when all available methods are used (Albattah *et al.* 2022). Cultural operations are aimed at preventing disease or impeding its occurrence. Common tomato diseases include: black mold, rot, crown and root rot, spotted wilt virus, radial rings cankers, mildews, blights, and many more (Ates *et al.* 2019, Gatahi 2020). Black mold fungus, *Alternaria alternate,* is one of the most common fungi experienced. It breeds dead organic matter whenever moisture is present and can be found on fruit and fermenting leaves in tomato fields before the fruit ripens (Rashid and Shoala 2020). *Alternaria* fungi is also responsible for early blight, and stem canker, which are often confused with black mold (Sánchez *et al.* 2022). These two diseases are caused by *A. solani* and *A. alternata* f.s. It is an economically destructive disease on tomato plants (Nazari *et al.* 2022). Controlling the most prevalent type of Fusarium crown and root-rot disease is very challenging because it is a soil-borne disease that economically restricts the production of greenhouse tomatoes. Due to the spread of microconidia, especially in greenhouses, FORL causes recurrent infections throughout the growing season and can cause up to 90% crop losses in greenhouse tomato cultivation. (Özbay *et al.* 2002). Commercially viable, agent-resistant indigenous varieties with adequate resistance to Fusarium crown and root-rot induced by FORL have not yet been developed (Ozbay and Newman 2004). A Gram-positive actinobacterium named *Clavibacter michiganensis* causes bacterial canker of tomatoes, a disease affecting millions of people worldwide (Peritore-Galve *et al.* 2020). Due to its rapid migration through the plant's vascular system and its ability to induce systemic symptoms, bactericidal agents are insufficient to manage this disease

Image processing and computer vision is key for detecting plant diseases (Ouhami *et al.* 2021). It is a technique that gathers plant images using computer vision tools and uses those images to determine whether or not pests and diseases are present (Vishnoi *et al.* 2021). Deep learning has recently been used in the evaluation and identification of plant diseases, facilitating early disease detection and diagnosis, thereby quickening the development of novel plant disease technology (Thakur *et al.* 2023). Deep learning offers very innovative methods to aid in the identification of diseases because of their computing capacity. For image-based research, CNN models are frequently used. They are effective at extracting high-level features from images (Ma *et al.* 2023). This study focused on detecting tomato fruit disease by proposing a 15-layer CNN architecture as a base network for single shot detector (SSD) to extract high features from tomato fruit. The article is organized as follows: the related work of this research, followed by the materials and methods used for this study, the proposed backbone, and finally, the experimental setup, results and discussion.

Materials and Methods

The proposed model architecture, hardware, dataset, and software resources used for this research are briefly discussed in this section. A single short shot detector with a proposed CNN base network was used in this study for tomato fruit disease detection. SSD was proposed to accurately locate the infected area of the tomato fruit. The experiments were carried out on a Lenovo laptop with an Intel Core i7 2.50 GHz processor and NVidia GeForce GTX 860M GPU.

Data collection and pre-processing

Tomato fruit images were collected from the internet and screened carefully to correspond with the disease classes. This included black mold diseases, radial ring diseases, spotted wilt diseases, and healthy tomato fruits. A training set and a testing set were created from the dataset in the following proportions: 8 : 2. Figure 1 shows samples of the tomato fruit classes in the dataset.

A total of 2500 images were obtained and a total of 650 images were obtained for the radial ring class representing 26% of the total data. Seven hundred images were obtained for the healthy class representing 28% of the total data, 600 images for the spotted wilt class representing 24% of the total data, and 550 images for the black mold class representing 22% of the total class. The data distribution is graphically presented in Figure 2.

To make the tomato fruit detectable by SSD, the MATLAB Image labeler app was used to annotate the datasets. The app offers a simple method for creating interactively a variety of shapes to classify as regions of interest (ROI). A rectangle shape was used to mark the disease spot's size and shape on the tomato fruit. Figure 3 shows sample annotated data in the MATLAB image labeler.

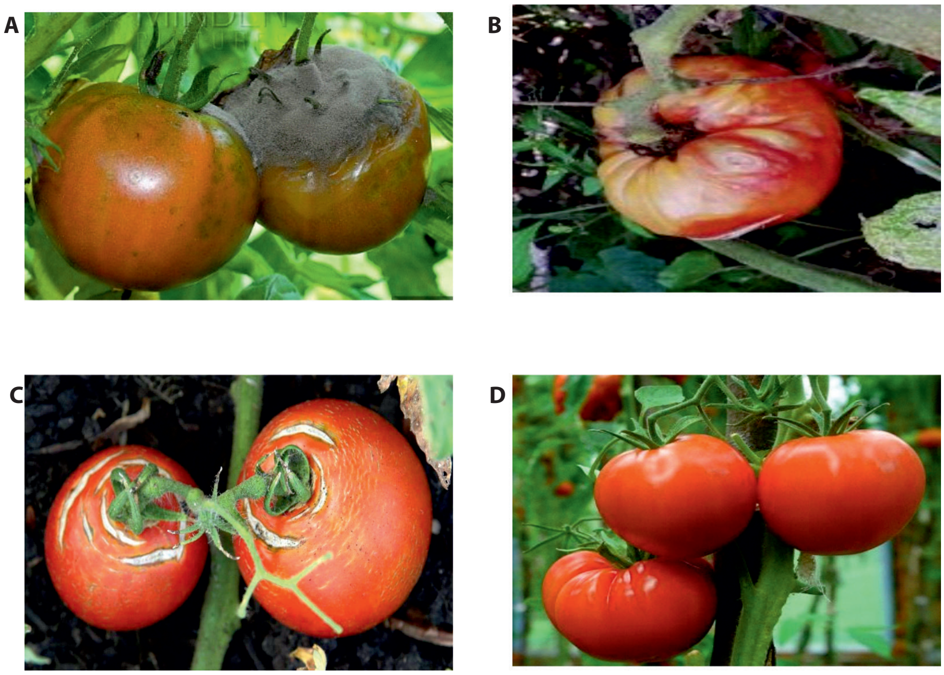

Fig. 1. Sample tomato images: A – black mold disease; B – spotted wilt virus; C – radial ring disease; D – Healthy tomato fruit

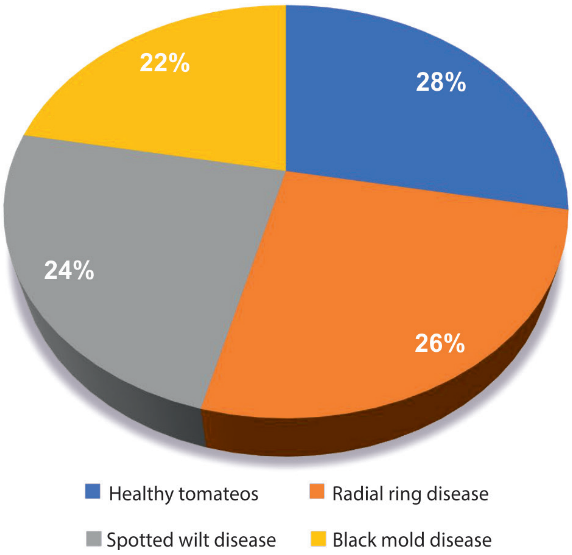

Fig. 2. Graphical presentation of dataset

Convolutional neural network

CNNs (Haar *et al.* 2023), or convolutional neural networks, have a complicated network topology and are capable of performing convolution operations. There are two broad categories of CNN-based image detection systems. In the first category, samples consisting of several candidate frames are created and then classified using CNN models. Several domains have successfully used deep neural networks. For the best possible image detection, CNN was used (Nyarko *et al.* 2022).

The input layer, convolution layer, pooling layer, full connection layer, and output layer make up the convolutional neural network design (Ji *et al.* 2022). One model repeatedly alternates between the convolution layer and the pooling layer, and when the neurons of the convolution layer are connected to the neurons of the pooling layer, full links are not necessary. In image identification, CNN is frequently used because of its magnificent potential (Gaba *et al.* 2022). CNNs have been used in the framework to manage the extracted image data that was gathered during the feature extraction stage (Ghazal 2022). The distinctive features of the affected tomato fruit are accurately extracted by the feature extraction layer of the proposed neural networks to achieve high recognition rates for tomato diseases (Bouni *et al.* 2023; Thakur *et al.* 2023). The performance of the proposed model was compared with the existing state-of-art models such as ResNet-50, AlexNet, and VGG19. Equation 1 illustrates a CNN layer-by-layer pass operation:

$$\rightarrow w^1 \rightarrow x^2 \rightarrow \ldots \rightarrow x^{n-1} \rightarrow w^{n-1} \rightarrow x^n \rightarrow w^n \rightarrow z, \tag{1}$$

where: x^1 – represents the input image, w^1 – denotes the parameter in the first layer, x^2– denotes the output of the first layer, and x^n – denotes the output of the final CNN layer. A loss layer is the final layer. The difference between the CNN prediction x^n and the target can be measured using a loss function, assuming that y is the corresponding target value for the input x^1. For instance, Equation 2 represents a straightforward loss function that could be:

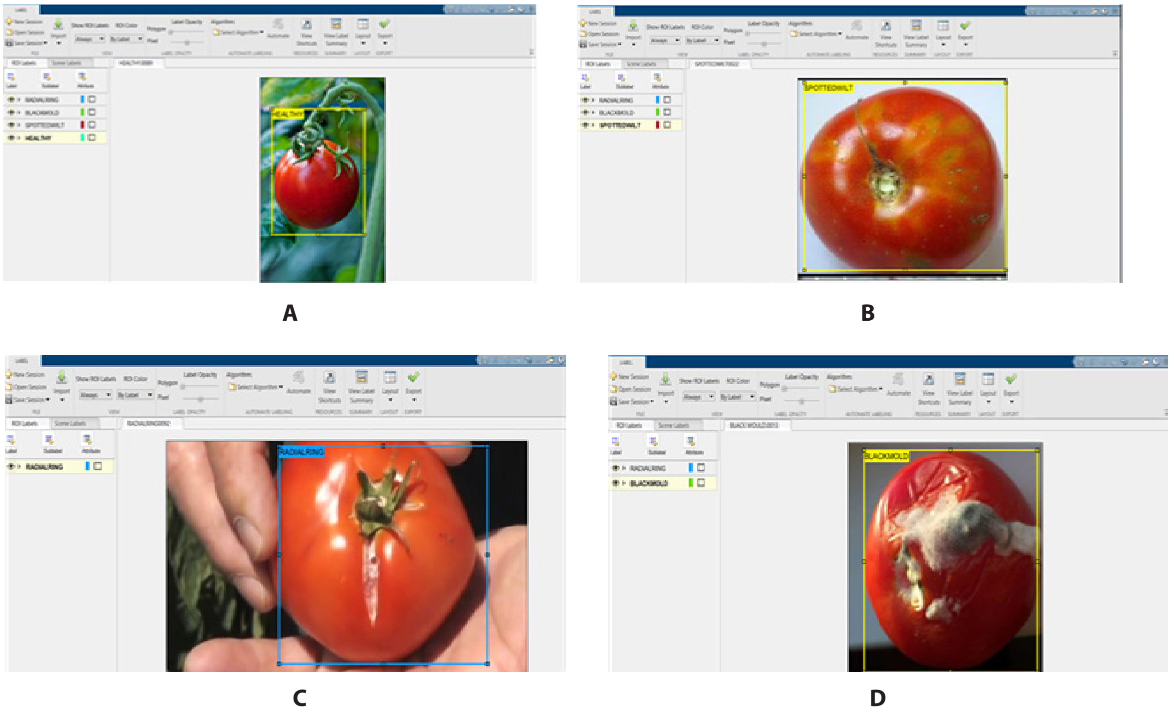

Fig. 3. Samples of annotated data in MATLAB image labeler. (A) Sample annotated healthy tomato in MATLAB image labeler. (B) Sample Annotated Spotted Wilt infected tomato in MATLAB image labeler. (C) Sample annotated radial ring affected Tomato in MATLAB image labeler. (D) A sample annotated black mold infected tomato in MATLAB image labeler.

$$z = 1/2\left\| y - x^n \right\|^2, \tag{2}$$

where: y – denote the ground truth value for the input, and x^n – denote the difference between the CNN prediction. The chain rule and vector calculus are essential to the CNN learning process. If $y \in \mathbb{R}^B$ is a scalar the partial derivative of a to y is a vector if a is a function of y, and it is defined in Equation 3 as:

$$\left[\frac{\partial a}{\partial y}\right] = \frac{\partial a}{\partial y_i}, \tag{3}$$

The element of $\frac{\partial a}{\partial y}$, a vector of the same size as y, is $\frac{\partial a}{\partial y_i}$. Given that $\left(\frac{\partial a}{\partial y^T}\right) = \left(\frac{\partial a}{\partial y}\right)^T$ and y is a function of x given that $x \in \mathbb{R}^W$ is another vector. The partial derivative of y concerning x is then expressed in Equation 4 as:

$$\left[\frac{\partial y}{\partial x^T}\right]_{ij} = \frac{\partial yi}{\partial xj}. \tag{4}$$

The H × D matrix that makes up this partial derivative has an entry at the point where the ith row and the ith column meet is $\frac{\partial yi}{\partial xj}$. In a chain-like argument, it is simple to see that a is a function of x: one function maps x to y, and another function maps y to a. One can compute using the chain rule in Equation 5:

$$\frac{\partial y}{\partial x^T} = \frac{\partial a}{\partial y^T}\frac{\partial y}{\partial x^T}. \tag{5}$$

Single shot detection algorithm (SSD)

In a single pass, the SSD model identifies objects with incredible detection accuracy, which saves a significant amount of time (Li *et al.* 2023, Vig *et al.* 2023). It yields accurate predictions at diverse levels from feature maps and achieves high detection accuracy by directly dividing predictions by aspect ratio (Guravaiah *et al.* 2023). Even on input images with low resolutions, these techniques produce high accuracy and simple end-to-end training. The SSD works by employing convolutional networks to produce a large number of bounding boxes of various fixed sizes and assess whether or not an instance of an object class is present in each box. After the convolutional networks have finished their job, a non-maximum suppression step then produces the final detections (Shi *et al.* 2022).

The head and the backbone make up the single shot detection design. The backbone layer is utilized by the feature map generator (Wei *et al.* 2023), a standard image classification network that has been trained. The final image classification layer produced by the model is removed, leaving the extracted feature maps. The SSD head is constructed by stacking convolutional layers and is placed on top of the backbone model (Liu *et al.* 2023a). The SSD algorithm divides each input image into grids of different sizes, and at each grid, detection for various classes and aspect ratios is performed. A score is assigned to each of these grids to represent how well an object works in that particular grid. The final detection is then extracted using non-maximum suppression from the collection of overlapping detections (Liu *et al.* 2023b). The SSD model's architecture is presented in Figure 4.

Proposed backbone (CNN)

The most significant area of research in the field of deep learning has always been the design of a high-quality, high-efficiency expressive network architecture. The majority of current network design approaches concentrate on how to combine features extracted from various layers and how to create computing units that can extract these features efficiently, increasing the expressiveness of the network. A key design section of a one-stage detector model is the backbone, which determines the quality of image feature extraction. It also affects the seceding object detection, recognition, and object classification. Figure 5 shows the proposed CNN architecture and Figure 6 shows the structure of the proposed backbone with a single shot detector algorithm.

The basic components of the proposed CNN used as the backbone for SSD in detecting tomato fruit diseases are described as follows

Input layer

It acts as an input element for the neural network. Each input layer feature passes its assigned value to each neuron in the first hidden layer in a top-to-bottom sequence. The hidden layer neuron then adds each value to the weight vector that corresponds to it, sums the multiplied values, applies its activation function to this total, and passes the value calculated by the activation function to the following layer. Input color channels are used to encrypt images. Each color level in the color channel at a specific location represents image data. RGB, which means red, blue, and green, is the most popular. The information contained in the image is the intensity of each channel color across the width and height of the input image. The blend of these three colors forms a color pattern.

Convolutional layer

The convolutional layer examines each neural network patch to derive more abstract properties. It performs convolution on the input using kernels, which were filters in the past. Convolution appears to be dot products between the filtering and the region they pass through

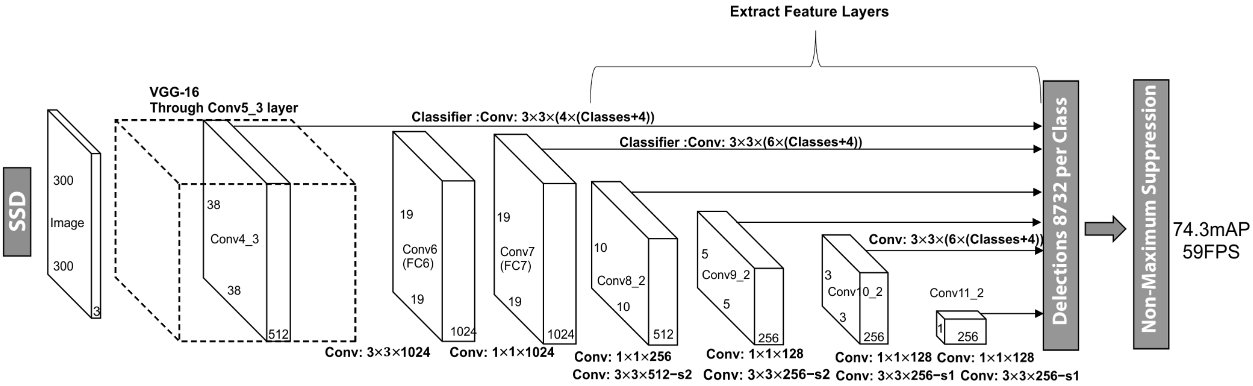

Fig. 4. Structure of single shot detector (SSD) architecture

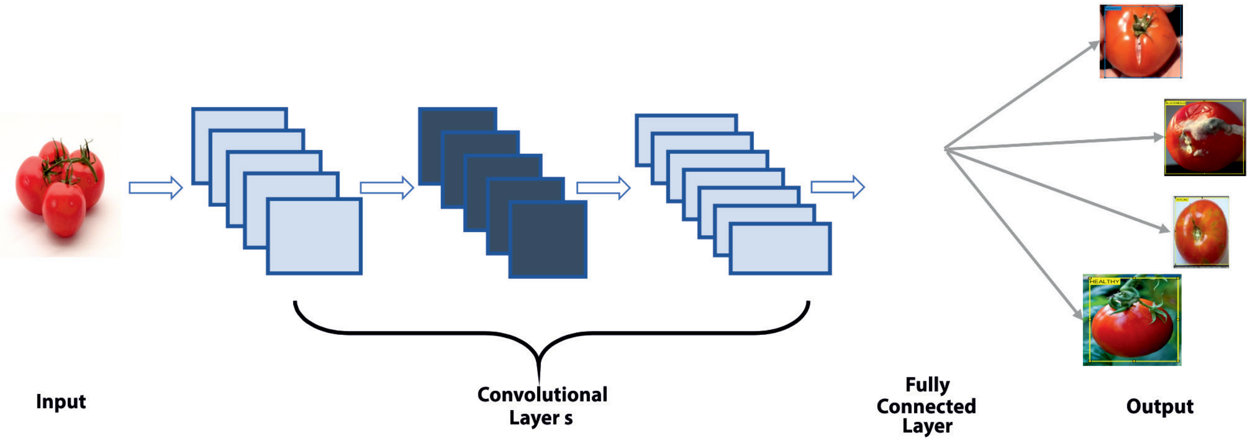

Fig. 5. Proposed CNN architecture

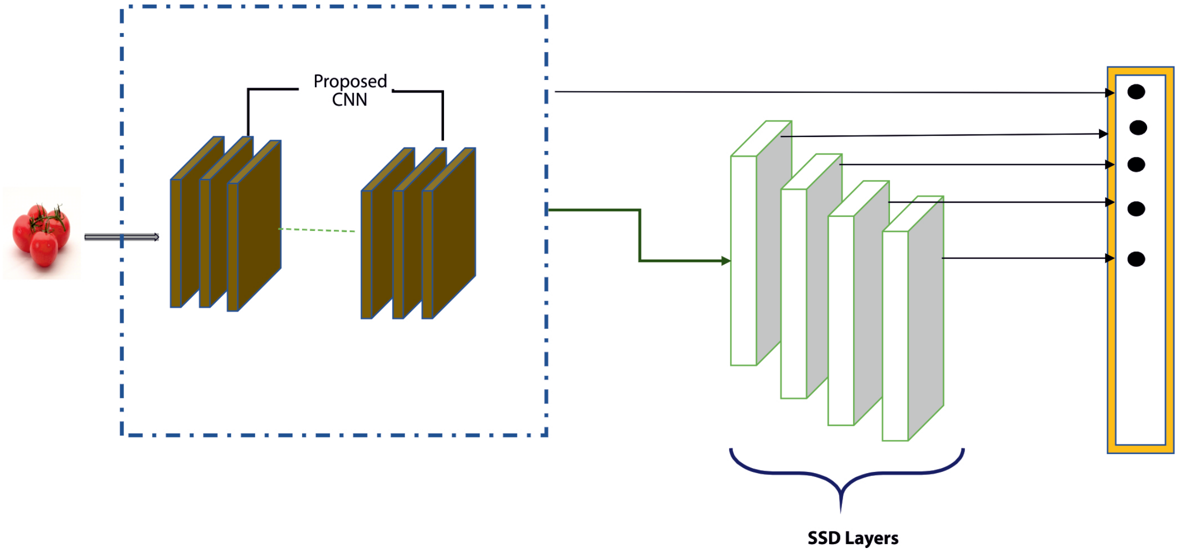

Fig. 6. Structure of proposed backbone with single shot detector (SSD)

in reality. To reduce the variable value and extract the key traits, a convolution layer was used. Rotation interpretability, scale interpretability, and interpretation interpretability were all included in the convolution layer. Both the generalization concept and the over-fitting issue were added to the fundamental framework. The operation of the convolution layer in CNN can be expressed in Equation 6 as follows:

$$y_{i^{l+1},j^{l+1},d} = \sum_{i=0}^{H} \sum_{j=0}^{B} \sum_{d^l}^{D^l} f_{i,d^l,} \text{ X } x^l_{i^{l+1},j^{l+1},d}, \quad (6)$$

$0 \leq$ d, $<$ D $= D^{l+1}$ for $i^{l+1,} j^{l+1}$ satisfying $0 \leq i^{l+1} < H^l - H + 1$, $< j^{l+1} < B^l - B + 1 = B^{l+1}$ in this equation, $x^l_{i^{i+1},d^l}$ refers to the element of x^l indexed $x^{l+1} + j^{l+1} + j, d^l$.

ReLu layer

A multi-layer neural network's ReLu layer is a nonlinear activation function. In this layer, all weak values, and the processed image are eliminated and switched out for zeros. Only when the node's input exceeds a predetermined threshold this feature is activated. The output will be zero if the input is less than zero. However, the input becomes linearly related to the dependent variable once it exceeds a certain threshold. This implies that the deep neural network's training data set can be processed more quickly than other activation functions. This prevents the sum from reaching zero. The ReLu layer can be considered as independent filtering for each element in the input: The input size is unchanged by a ReLu layer. ReLu layers maintain the input's original size, so/and y have the same size. This is expressed in Equation 7 as:

$$y_{i,j,d} = \max\{0, x^l_{i,j,d}\}, \quad (7)$$

$$0 < i < H^{l+1},\ 0 < j < Bl = B^{l+1n} \text{ and } 0 \leq d \leq D^l \leq D^{l+1}.$$

ReLu layers do not carry any parameters, so this layer does not require parameter learning. From Equation 7 we obtain Equation 8 as:

$$\frac{dy_{i,j,d}}{dx^l_{i,j,d}} = \left\| x^l_{i,j,d} > 0 \right\|, \quad (8)$$

where $\|.\|$ – the indicator function; it returns a value of 1 if its argument is true and a value of 0 otherwise. Equation 9 is obtained as:

$$\left[\frac{\partial a}{2x^l}\right]_{i,j,d} = \begin{cases} \left[\frac{\partial a}{2x^l}\right]_{i,j,d} & \text{if } x^l_{i,j,d} > 0, \\ 0 \end{cases} \quad (9)$$

where: y – the alias for x^{l+1}. ReLu's objective is to make the CNN more nonlinear. Since the semantic content of an image (in this case, a tomato fruit) is a highly nonlinear mapping of input pixel values.

Batch normalization layer

Normalization of the inputs to a layer is done in tiny batches using a deep-learning training technique known as batch normalization. Training is accelerated by the batch normalization layer, which also lessens the impact of initialization after the convolution operation. This layer is added before the input of each convolutional layer to ensure that each layer's input has the same distributions and to reduce inner covariate shifts during training. Therefore, the process of learning is regulated and a deep network can be trained with significantly fewer training epochs.

MaxPooling layer

Two input arguments are required by a MaxPool layer, height, and core width. The kernel moves across the pixels in a straight line at the predetermined step size, beginning in the upper left corner of the feature map. The kernel window's highest-valued pixel serves as the source of the value for the associated node in the pooling layer. Max CNNs' pooling layers are a crucial component. They keep the number of network parameters to a minimum while summarizing the activation maps, the result of pooling (y, or x in its place), ($H \times B \times D$) will be the size of $H^{l+1} + B^{l+1} + D^{l+1}$ in order 3-tensor as seen in Equation 10:

$$H^{l+1} = \frac{H^l}{H}, B^{l+1} \; = \frac{B^l}{B}, \; D^{l+1} = \; D^l \quad (10)$$

where: H^{l+1}, and D^{l+1} – the input to the pooling layer. A max-pooling operator maps a subregion to its maximum value. Using exact mathematics in Equation 11 as:

$$\text{max: } y_{i^{l+1}, j^{l+1}, d} = \begin{matrix} max \\ 0 < i < H, 0 < j < B \end{matrix} \; x^l_{i^{l+1}} \quad (11)$$

$$\times H + i, j^{l+1} \times B + j, d.$$

where: $0 \leq i^{l+1} < H^{l+1}$, $0 \leq j^{l+1} < B^{l+1}$ and $0 \leq d < D^{l+1}$ – depicts a pooling local operator with straight forward computation.

Fully connected layer

The flat layer, a two-dimensional (2d) layer, provides input to a completely connected layer. The affine function receives data from the smoothing layer before passing it on to the nonlinear function. One FC (fully connected) is the result of combining one affine function with one nonlinear function. The suggested architecture's final probability for each label is provided in this layer. The FC layer transforms incoming neurons using weighted linear transformations and sends outputs to nonlinear neurons. Every neuron in one layer is connected to every neuron in the other layers, forming a succession of fully connected layers that make up a fully connected neural network. Fully connected networks have the major benefit of being "structure-agnostic", meaning that no specific assumptions about the input are required. The descriptions of mathematical models with completely connected layers are provided in Equation 12. Considering a convolution or pooling layer/node with the dimensions:

$$a_j^{[i]} = \; \Sigma_{(l-1)}^{n(i-l)} \, B_j^i \, 1 B_l^{(i-1)} + \; d_j^{[l]} \; \rightarrow \; a_j^{[i]} = \varphi^{[i]} \left(z_j^{[i]} \right). \quad (12)$$

The input $z_j^{[i]}$ is the output of the pooling operation with $H^{l+1} + B^{l+1} + D^{l+1}$.

Softmax layer

It is mainly used to represent certainty probabilities in neural network outputs by scaling output between 0 and 1. It is possible to compute normalization by dividing the output under the study's exp value by the sum of all possible outputs' exp values. The softmax function transforms a vector of k real values into a vector of k real values with a total of 1. Softmax converts input values, which can be either positive or negative, zero or higher than 1, into values between 0 and 1, making them understandable as probabilities. Any small or negative, input is transformed into a small chance by Softmax, and any large input is transformed into a large probability.

Output layer

The neural network's output layer is the final layer that produces the desired predictions. An output layer in a neural network generates the desired outcome prediction. Before determining the final output, it applies its own set of weights and biases. For some issues, the hidden layer activation function may be different from the output layer activation function. For instance, in classification issues, the final classes are derived using softmax activations. A summary of the proposed CNN Architecture is given in Figure 7.

Feature extraction

Different subsections of the feature extraction phase, such as color extraction, texture extraction, and infected area extraction were considered. An RGB image, in terms of conventional color space, is a mixture of the colors red, green, and blue. Therefore, it is appropriate to determine the proportion of red, green, and blue components in each image pixel. One of the main components for detecting tomato disease is color extraction. This technique takes a sample image of a tomato and extracts a spectrum of colors from lightest to darkest. Entropy calculation is used to perform texture detection on segmented captured images. This technique first converts the segmented image to grayscale. Following grayscale conversion, the system uses a conventional entropy calculation method to extract the local entropy of the image. The system determines the minimum and maximum pixels present in the local entropy matrix following its extraction. The texture quality coefficient will be equal to the mean of the calculated minimum and maximum pixel range from the local entropy matrix. A rough idea of tomato texture can be obtained from the texture quality coefficient calculated from the local entropy matrix. The created system is trained using an extensive set of texture quality coefficients that were extracted from various tomato sample images in the created dataset as:

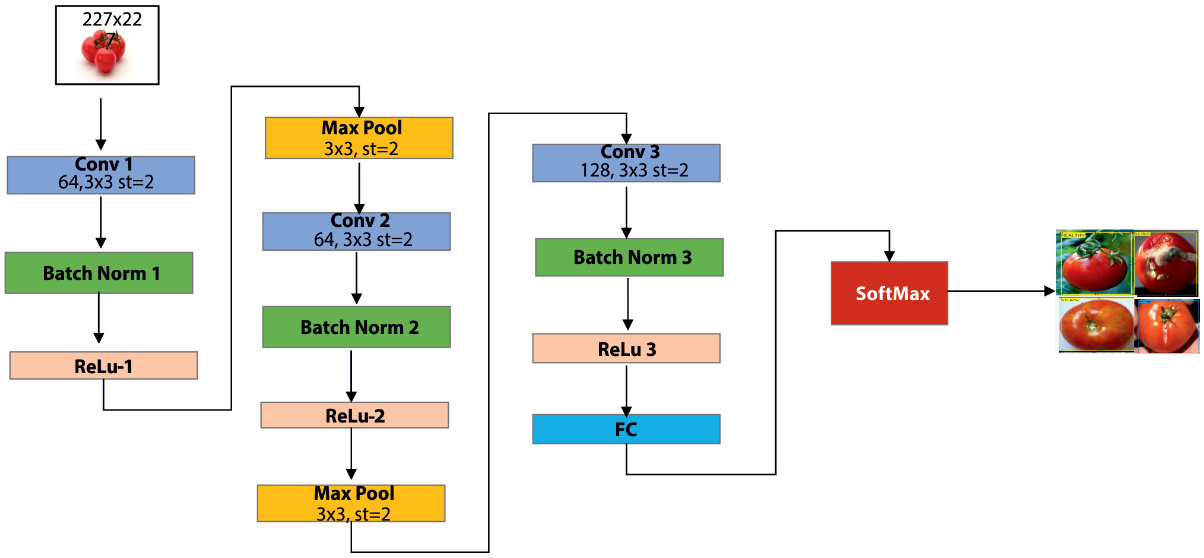

Fig. 7. Summary of the proposed backbone structure

$$A(x,y,z) = \sum_{i=1}^{i=n} \sum_{i=1}^{j=n} \sum_{k=1}^{n} B(i,j,k) \times \log 2B(i,j,k),$$

$$B(x,y,z) = \sum_{i=1}^{i=n} \sum_{i=1}^{j=n} \sum_{k=1}^{n} {}^{n\,(i,j,k)}/_{m^2}, \tag{13}$$

where: $B(x,y,z)$ – the calculated probability index matrix, and m^2 – the size of the segmented image.

Results, Discussion and Conclusions

The specifics of the experiments including parameter fitting and the outcomes of the experiments are covered in this section. Each model's performance is clearly presented.

Quantitative Analysis

The primary indicators for evaluating a deep learning model's detection performance are precision, average precision, accuracy, and recall. The training time and image detection time were used to obtain the mean average precision (mAP) of each model's performance in this study.

Precision

Precision is the ratio of the number of positively detected samples that were correctly identified as positive to the number of positively detected samples. Precision is defined in Equation 14 as:

$$Precision = {}^{Tp}/_{Tp + Fp}, \tag{14}$$

where: Tp – the number of positive samples that were correctly classified, that is, the real positive samples that the classifier also correctly classified as positive samples, and Fp – the number of negative samples that were incorrectly classified as positive, that is, the real negative samples that the classifier incorrectly classified as positive samples.

Mean average precision (mAP)

To determine the mAP, the mean average precision compares the detected box to the ground-truth bounding box. When the ratings are high, the model's detections are more accurate. Average precision is defined for datasets with multiple classes as in Equation 15 as:

$$mAP = {}^{\sum Average\ Precsion}/_{N'}, \tag{15}$$

where: Σ – represents the sum of average precision and N'– represents the total number of all classes in the dataset.

Accuracy

Accuracy, or the percentage of correctly detected images in all instances, is a term used to describe the rate at which images are successfully detected. Additionally, the proportion of samples correctly identified for a given test database to the total number of data successfully classified by the predictor is proportional to the total amount of data. Accuracy is described in Equation 16:

$$Accuracy = {}^{Tp + Tn}\big/_{Tp + Fp + Tn + Fn}, \quad (16)$$

where: *Tp* – the number of positive samples that were correctly classified, *Fp* – the number of negative samples that were incorrect, *Tn* – negative classes that were correctly predicted as negative, and *Fn* – positive classes that were falsely predicted as negative.

Root mean square error (RMSE)

The average distance between the values in the dataset and those predicted by the model. A given model can "fit" a dataset more accurately when the RMSE is low. The root means square error or RMSE is calculated using the following formula:

$$RMSE = \sqrt{\sum {}^{(P_i - O_i)^2}\big/_{n}}, \quad (17)$$

where: Σ – the sum of RMSE, *P*i is the predicted value for the i^{th} – observation in the dataset, *Oi* – the observed value for the observation in the dataset, and is the sample size.

Intersection over union (IOU)

A metric quantifies how well the predicted and actual boxes match. By dividing the area of the intersection between the two boxes by the area of their union, the IoU is determined using the following equation. The accuracy of the prediction increases with IoU.

$$IOU = {}^{Intersection\ Area}\big/_{Union\ Area}. \quad (18)$$

A novel convolutional neural network method was proposed in this study for tomato disease detection. The proposed model serves as a backbone for the single shot object detector algorithm. The main aim of this research was to improve the accuracy of the existing deep learning models in plant disease detection. The total epochs for training and testing were 100 epochs at 1000 iterations. Seventy percent of the total datasets was utilized for training, and 30% was used for data testing. Table 1 shows the results of the proposed CNN with SDD in detecting tomato fruit disease.

The outcome of the experimental results showed that the proposed CNN backbone with single shot detector algorithm achieved excellent results of 0.991 precision, 0.994 mean average precision, 0.988 accuracy, and a root mean square error of 1.21 The performance evaluation of the proposed CNN backbone with SSD Algorithm is presented in Table 1.

The accuracy, precision, mean average precision (mAP), intersection over union (IOU), and root mean square error (RMSE), at 1000th for the proposed model were recorded. The performance of the proposed model was compared with existing models like ResNet-50, AlexNet, VGG16, and VGG19. The model's performance is shown in Table 2.

The classes in the dataset included black mold diseases, radial ring diseases, spotted wilt diseases, spotted wilt viruses, and healthy tomato fruits. The performance of the proposed model in each class was evaluated and the outcome is seen in Table 3. The results of the performance evaluation show that the proposed model obtained higher accuracy, precision, and mean average precision on healthy tomato fruit than black mold, spotted wilt, and radial ring.

Figure 8 shows sample test results from the experiments. The results show the class name/ the disease of the tomato fruit and the percentage of the detection results. Figure 9 shows the train accuracy and loss graphs of the models used in this study.

To identify the model that was the most effective at detecting tomato fruit diseases accuracy, mean average precision, precision, recall, and root mean square error score values were used as evaluation metrics to assess

Table 1. Performance evaluation of the proposed backbone-SSD

Metrics	Results
Precision	0.991
Mean average precision (mAP)	0.994
Accuracy	0.988
Recall	0.991
Root mean square error	1.21

Table 2. Models performance evaluation on test data

Models	Accuracy	Precision	mAP	Recall	RMSE
SSD-ResNet-50	0.985	0.990	0.995	0.991	1.27
SSD-AlexNet	0.963	0.953	0.952	0.981	1.28
SSD-VGG-16	0.963	0.951	0.953	0.97	1.28
SSD-VGG-19	0.966	0.954	0.962	0.98	1.28
SSD-Proposed CNN	0.988	0.991	0.994	0.991	1.21

mAP – mean average precision

Table 3. Proposed model performance in each class

Class	Accuracy	Precision	mAP
Healthy	0.985	0.985	0.982
Black mold	0.975	0.974	0.971
Spotted wilt	0.953	0.951	0.950
Radial ring	0.967	0.966	0.961

mAP – mean average precision

the performance of the pre-trained models. A graph showing the validation accuracy for the pre-trained models was created using the validation accuracy calculated in Figure 9.

Discussions

This section presents a thorough analysis of recent research on plant disease detection via deep learning methods. Ignoring the early indicators of plant disease in the agricultural sector can result in losses in food harvests and ultimately lead to the collapse of the global food industry. In a few selected diseases and crops, the recent trend of using different machine-learning algorithms for plant disease detection has yielded encouraging results.

A crop conditional model was developed which utilized a unique CNN design in conjunction with crop metadata to recognize 17 diverse diseases in five crops. These crops included rapeseed, barley, rice, wheat, and corn. From the obtained robust features of a large-size multi-crop dataset, the model quickly learned similar disease symptoms in different crops, which decreased the complexity of the classification function (Picon *et al.* 2019)

Durmuş *et al.* (2017) proposed a DL method for tomato leaf disease detection. They aimed at using a robot to run real-time plant leaf detection manually or autonomously in the field or greenhouse. Training and validation were done by adopting AlexNet and Squeeze models on the plant image dataset. The examined tomato leaf diseases in their research cause a physical change. RGB cameras can observe these alterations in the leaves. The output of their novel approach showed that the AlexNet model performed slightly better than the SqueezeNet model. They concluded that the SqueezeNet model is 80 times smaller than the AlexNet Model, and the cause of the differences was attributed to the Caffe format (Durmuş *et al.* 2017).

(Iqbal *et al.* 2021) proposed a gray level co-occurrence matrix (GLCM) algorithm to calculate 13 distinctive statistical features of tomato leaves. To categorize data, the support vector machine (SVM)

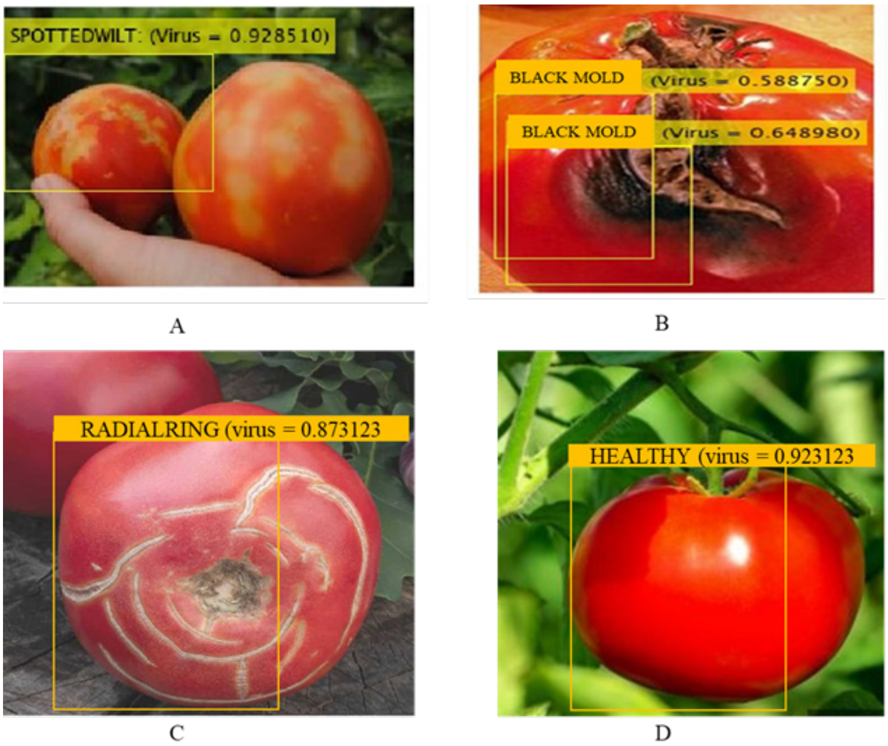
SPOTTEDWILT: (Virus = 0.928510)
A
BLACK MOLD (Virus = 0.588750)
BLACK MOLD (Virus = 0.648980)
B
RADIALRING (virus = 0.873123
C
HEALTHY (virus = 0.923123
D

Fig. 8. Sample test results. (A) Sample detected Spotted wilt virus at 0.92%. (B) Sample detected black mold virus at 0.58 and 0.64 respectively. (C) Sample detected radial ring virus at 0.87%. (D) Sample detected Healthy tomato fruit at 0.92%.

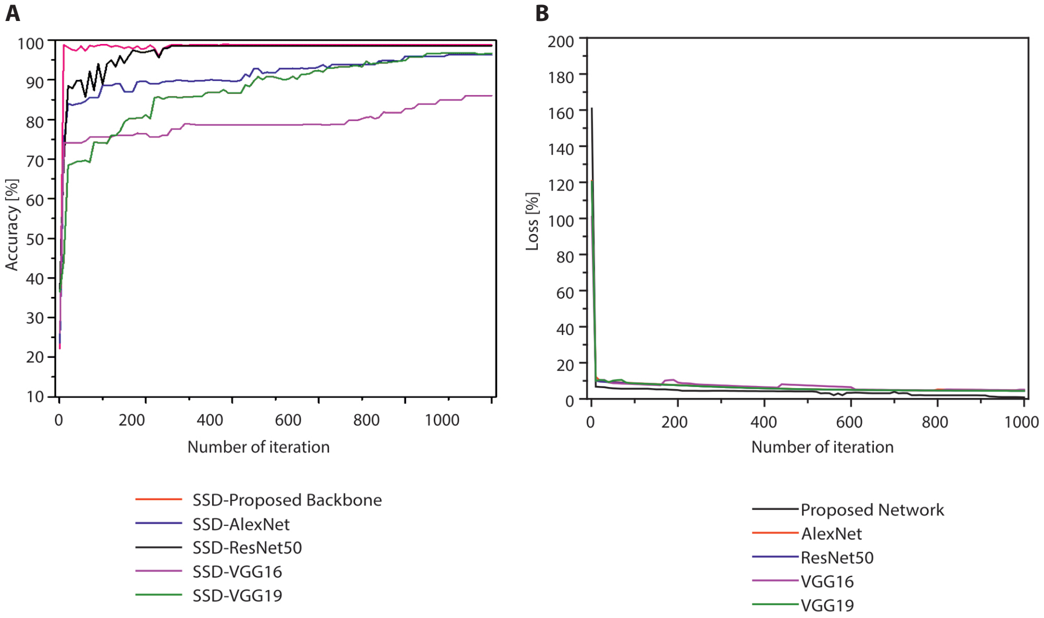

Fig. 9. Training-accuracy graphs for the models – A; Training-loss graphs for the models – B

was utilized. The features obtained from the GLCM algorithm were implemented as a mobile application. The processed leaf was compared with the features stored to recognize the tomato leaf disease. The experimental findings of their method provide 100% accuracy for healthy leaves, 95% for early blight, 90% for Septoria leaf, and 85% for late blight (Iqbal *et al.* 2021).

For effective plant disease identification, pre-trained models based on convolutional neural networks (CNN) were used. The focus was on fine-tuning the hyperparameters of well-known pre-trained models like DenseNet-121, ResNet-50, VGG-16, and Inception V4, in particular. The experiments used the well-known Plant Village dataset, which contains 54,305 images of various plant disease species organized into 38 classes. Through classification accuracy, sensitivity, specificity, and F1 score, the model's performance was assessed. Additionally, a comparison with comparable cutting-edge studies was done. The tests showed that DenseNet-121 outperformed cutting-edge models by 99.81% in terms of classification accuracy (Andrew *et al.* 2022).

To identify plant species in images, most of the studies reviewed adopted pre-trained CNN models by fine-tuning, conducting computational statistics of the tomato leaf features, and using a robot in detecting tomato plant disease in the greenhouse. The performance of their methods was compared with other state-of-the-art models. Their findings showed that deep convolutional neural networks perform well in the identification of plant diseases. However, more work is required to improve the previous studies, such as proposing crop disease models that can extract higher features of the affected area of the crop to enable accurate disease detection. A 15-layer convolutional neural network was proposed as the backbone of a single short shot detector model to address these challenges. This method can identify the affected region and the shapes of the infected areas.

This research focused on improving the single shot detecting algorithm by proposing a CNN backbone for tomato fruit disease detection. The main aim of this research was to identify the common disease that affects the tomato fruit, gather a tomato fruit dataset, and enhance the deep learning framework for detecting plant disease. The proposed method forecasts and creates a new prototype that will offer improved plant disease detection performance with less computational resources in a short time. The results obtained from the proposed method performed better than SSD with ResNet-50, AlexNet, VGG16, and VGG19 backbones in tomato fruit disease detection. In a subsequent study, we will concentrate on how to enhance the tomato disease dataset to befit general target detection algorithms.

The detection and management of crop and plant infestations have been significantly improved by deep learning technologies. Complex disease and pest identification has become possible due to advanced developments of object detectors. Large and real-time datasets are supported, automatic feature extraction functionality is also available, and the overall execution time is decreased. Therefore, deep learning techniques can be taken into account for upcoming research in the agriculture sector, such as roots, land, weeds, leaves, and fruits for disease identification.

However, the majority of this research is lab-based and heavily depends on gathered images of plant diseases and pests. To increase the robustness and generalization of the deep learning models, future research will focus on collecting images from various plant growth stages, seasons, and geographic locations.

Early detection of plant diseases and pests is essential for halting and controlling their growth., Accurate identification and prediction require the incorporation of meteorological and plant health data, such as temperature and humidity. Deep learning model training and network learning can benefit from unsupervised learning as well as combining prior knowledge of brain-like computers with human visual cognition. Collaboration between experts in agriculture and plant protection in future research is necessary to realize the full potential of this technology. Their skills and knowledge must be combined with deep learning algorithms and models, and the resulting information will be incorporated into farming machinery.

References

Albattah W., Javed A., Nawaz M., Masood M., Albahli S. 2022. Artificial Intelligence-based drone system for multiclass plant disease detection using an improved efficient convolutional neural network. Frontiers in Plant Science 13: 1003152. DOI: https://doi.org/10.3389/fpls.2022.808380.

Andrew J., Eunice J., Popescu D. E., Chowdary M.K., Hemanth J. 2022. Deep learning-based leaf disease detection in crops using images for agricultural applications. Agronomy 12 (10): 2395. DOI: https://doi.org/10.3390/agronomy12102395

Ates C., Fidan H., Karacaoglu M., Dasgan H. 2019. The identification of the resistance levels of *Fusarium oxysporum f. sp. radicis-lycopersici* and tomato yellow leaf curl viruses in different tomato genotypes with traditional and molecular methods. Applied Ecology and Environmental Research 17 (2): 2203–2218. DOI: http://dx.doi.org/10.15666/aeer

Bhujel A., Kim N.-E., Arulmozhi E., Basak J., Kim H.-T. 2022. A lightweight attention-based convolutional neural networks for tomato leaf disease classification. Mdpi- Agriculture 12 (2): 228. DOI: https://doi.org/10.3390/agriculture12020228

Bouni M., Hssina B., Douzi K., Douzi S. 2023. Impact of pretrained deep neural networks for tomato leaf disease prediction. Journal of Electrical and Computer Engineering 2023 (1): 1–11. DOI: https://doi.org/10.1155/2023/5051005

Caruso A.G.S.B., Parrella G, Rizzo R., Davino S., Panno S. 2022. Tomato brown rugose fruit virus: a pathogen that is changing the tomato production worldwide. Annals of Applied Biology 181 (3): 258–274. DOI: https://doi.org/10.1111/aab.12788

Durmuş H., Güneş E. O., Kırcı M. 2017. Disease detection on the leaves of the tomato plants by using deep learning. p. 1–5. In: Proceedings of "6th International Conference on Agro-Geoinformatics". 7–10 August 2017, Fairfax, VA, USA. DOI: https://doi.org/10.1109/Agro-Geoinformatics.2017.8047016.

Gaba S., Budhiraja I., Kumar V., Garg S., Kaddoum G., Hassan M. M. 2022. A federated calibration scheme for convolutional neural networks: models, applications and challenges. Computer Communications 192: 144–162. DOI: https://doi.org/10.1016/j.comcom.2022.05.035

Gatahi D. 2020. Challenges and opportunities in tomato production chain and sustainable standards introduction. International Journal of Horticulture Science and Technology 7 (3): 235–262. DOI: https://doi.org/10.22059/ijhst.2020.300818.361

Ghazal T. 2022. Convolutional neural network based intelligent handwritten document recognition. Computers, Materials & Continua 70 (3): 4563–4581. DOI: https://doi.org/10.32604/cmc.2022.021102

Golan K., Kot I., Kmieć K., Górska-Drabik E. 2023. Approaches to integrated pest management in Orchards: *Comstockaspis perniciosa* (comstock) case study. Mdpi- Agriculture 13 (1): 131. DOI: https://doi.org/10.3390/agriculture13010131

Guravaiah K., Bhavadeesh Y.S., Shwejan P., Vardhan A.H., Lavanya S. 2023. Third eye: object recognition and speech generation for visually impaired. Procedia Computer Science 218: 1144–1155. DOI: https://doi.org/10.1016/j.procs.2023.01.093

Haar L. V., Elvira T., Ochoa O. 2023. An analysis of explainability methods for convolutional neural networks. Engineering Applications of Artificial Intelligence 117: 105606. DOI: https://doi.org/10.1016/j.engappai.2022.105606

Hemathilake D., Gunathilake D. 2022. Agricultural productivity and food supply to meet increased demands. Future Foods 2022: 539–553. DOI: https://doi.org/10.1016/B978-0-323-91001-9.00016-5

Hofman-Bergholm M. 2023. A transition towards a food and agricultural system that includes both food security and planetary health. Mdpi-Foods 12 (1): 12. DOI: https://doi.org/10.3390/foods12010012

Humbal A., Pathak B. 2023. Application of nanotechnology in plant growth and diseases management: tool for sustainable agriculture. p. 145–168. In: "Agricultural and Environmental Nanotechnology" (F-L. Fabian, J.K. Patra, eds.). Springer, Singapore, 674 pp. DOI: https://doi.org/10.1007/978-981-19-5454-2_6

Iqbal N., Mumtaz R., Shafi U., Zaidi S. M. H. 2021. Gray level co-occurrence matrix (GLCM) texture based crop classification using low altitude remote sensing platforms. PeerJ Computer Science 7 (8): e536. DOI: https://doi.org/10.7717/peerj-cs.536

Ji Y., Liu S., Hao Y. 2023. Realization of convolutional neural network based on FPGA. p. 761–765. In: Proceedings of "Third International Conference on Computer Vision and Data Mining (ICCVDM 2022). Hulun Buir, China. DOI: https://doi.org/10.1117/12.2660023

Khan H. R., Gillani Z., Jamal M.H., Athar A., Chaudhry M.T., Chao H., He Y., Chen M. 2023. Early identification of crop type for smallholder farming systems using deep learning on time-series sentinel-2 imagery. Mdpi-Sensors 23 (4): 1779. DOI: https://doi.org/10.3390/s23041779

Knapp S., Peralta I.E. 2016. The tomato (*Solanum lycopersicum* L., Solanaceae) and its botanical relatives. p. 7–21. In "The Tomato Genome" (M. Causse, J. Giovannoni, M. Bouzayen, Z. Mohamed, eds.). Springer Berlin, Heidelberg, 259 pp. DOI: https://doi.org/10.1007/978-3-662-53389-5_2

Kremneva O., Danilov R., Gasiyan K., Ponomarev A. 2023. Spore-trapping device: an efficient tool to manage fungal diseases in winter wheat crops. Plants 12 (2): 391. DOI: https://doi.org/10.3390/plants12020391

Leite G.L.D., Fialho A. 2018. Sustainable management of arthropod pests of tomato. p. 305–311. In: "Sustainable Management of Arthropod Pests of Tomato" (W. Wakil, G.E. Brust, T.M. Perring, eds.). Academic Press. DOI: https://doi.org/10.1016/B978-0-12-802441-6.00014-0

Li W., Zhang H., Wang G., Xiong G., Zhao M., Li G., Li R. 2023. Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing. Robotics and Computer-Integrated Manufacturing 80: 102470. DOI: https://doi.org/10.1016/j.rcim.2022.102470

Liu W., Liu K., Chen D., Zhang Z., Li B., El-Mogy M. M., Tian S., Chen T. 2022. *Solanum lycopersicum*, a model plant for the studies in developmental biology, stress biology and food science. Foods 11 (16): 2402. DOI: https://doi.org/10.3390/foods11162402

Liu B., Luo L., Wang J., Lu Q., Wei H., Zhang Y., Zhu W. 2023a. An improved lightweight network based on deep learning for grape recognition in unstructured environments. Information Processing in Agriculture 2023. DOI: https://doi.org/10.1016/j.inpa.2023.02.003 (in press)

Liu H., Wang D., Xu K., Zhou P., Zhou D. 2023b. Lightweight convolutional neural network for counting densely piled steel bars. Automation in Construction 146: 104692. DOI: https://doi.org/10.1016/j.autcon.2022.104692

Ma X., Man Q., Yang X., Dong P., Yang Z., Wu J., Liu C. 2023. Urban feature extraction within a complex urban area with an improved 3D-CNN using airborne hyperspectral data. Remote Sensing 15 (4): 992. DOI: https://doi.org/10.3390/rs15040992

Mohan A., Krishnan R., Arshinder K., Vandore J., Ramanathan U. 2023. Management of postharvest losses and wastages in the Indian tomato supply chain – a temperature-controlled storage perspective. Mdpi-Sustainability 15 (2): 1331. DOI: https://doi.org/10.3390/su15021331

Nazari K., Ebadi M. J., Berahmand K. 2022. Diagnosis of alternaria disease and leafminer pest on tomato leaves using image processing techniques. Journal of the Science of Food and Agriculture 102 (15): 6907–6920. DOI: https://doi.org/10.1002/jsfa.12052

Nkongho R.N., Ndam L.M., Akoneh N.N., Tongwa Q.M., Njilar R.M., Agbor D.T., Sama V., Ojongakpa O.T., Ngone A.M. 2023. Vegetative propagation of F1 tomato hybrid (*Solanum lycopersicum L.*) using different rooting media and stem-nodal cuttings. Journal of Agriculture and Food Research 11: 100470. DOI: https://doi.org/10.1016/j.jafr.2022.100470

Nyarko B.N.E., Bin W., Zhou J., Agordzo G.K., Odoom J., Koukoyi E. 2022. Comparative analysis of AlexNet, Resnet-50, and Inception-V3 models on masked face recognition. p. 337–343. In: Proceedings of "2022 IEEE World AI IoT Congress (AIIoT)". 6–9 June 2022. DOI: https://doi.org/10.1109/AIIoT54504.2022.9817327

Ouhami M., Hafiane A., Es-Saady Y., El Hajji M., Canals R. 2021. Computer vision, IoT and data fusion for crop disease detection using machine learning: a survey and ongoing research. Remote Sensing 13 (13): 2486. DOI: https://doi.org/10.3390/rs13132486

Ozbay N., Newman S.E. 2004. Fusarium crown and root rot of tomato and control methods. Plant Pathology Journal 3 (1): 9–18. DOI: https://doi.org/10.3923/ppj.2004.9.18

Özbay N., Newman S., Brown W. 2004. Evaluation of *Trichoderma harzianum* strains to control crown and root rot of greenhouse fresh market tomatoes. Acta Horticulturae 635: 635. DOI: https://doi.org/10.17660/ActaHortic.2004.635.10

Peritore-Galve C., Tancos M., Smart C. 2020. Bacterial canker of tomato: revisiting a global and economically damaging seedborne pathogen. Plant Disease 105 (6): 1581–1595. DOI: https://doi.org/10.1094/PDIS-08-20-1732-FE

Picon A., Seitz M., Alvarez-Gila A., Mohnke P., Ortiz-Barredo A., Echazarra J. 2019. Crop conditional convolutional neural networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions. Computers and Electronics in Agriculture 167: 105093. DOI: https://doi.org/10.1016/j.compag.2019.105093

Rashid I., Shoala T. 2020. Nanoactivities of natural nanomaterials rosmarinic acid, glycyrrhizic acid and glycyrrhizic acid ammonium salt against tomato phytopathogenic fungi *Alternaria alternata* and *Penicillium digitatum*. Journal of Plant Protection Research 60 (2): 1–11. DOI: https://doi.org/10.24425/jppr.2020.133309

Sánchez P., Vélez-del-Burgo A., Suñén E., Martínez J., Postigo I. 2022. Fungal allergen and mold allergy diagnosis: Role and relevance of *Alternaria alternata Alt* a 1 protein family. Journal of Fungi 8 (3): 277. DOI: https://doi.org/10.3390/jof8030277

Shi M., He P., Shi Y. 2022. Detecting extratropical cyclones of the northern hemisphere with single shot detector. Mdpi-Remote Sensing 14 (2): 254. DOI: https://doi.org/10.3390/rs14020254

Sreedevi A., Manike C. 2023. Development of weighted ensemble transfer learning for tomato leaf disease classification solving low resolution problems. The Imaging Science Journal 71 (2): 1–27. DOI: https://doi.org/10.1080/13682199.2023.2178605

Thakur P.S., Sheorey T., Ojha A. 2023. VGG-ICNN: A Lightweight CNN model for crop disease identification. Multimedia Tools and Applications 82 (1): 497–520. DOI: https://doi.org/10.1007/s11042-022-13144-z

Thakur R., Mohanty S., Sethy P.K., Patro N., Sethy P., Achary A.A. 2023. Detection of tomato leaf ailment using convolutional neural network technique. p. 193–202. In: Proceedings of "Third Mobile and Radio Communications and 5G Networks. 10–12 June, 2022. Krukshatra, India. DOI: https://doi.org/10.1007/978-981-19-7982-8_17

Vig S., Arora A., Arya G. 2023. Automated license plate detection and recognition using deep learning. p. 419–431 In: Proceedings of "Advancements in Interdisciplinary Research: First International Conference, AIR 2022". 6–7 May 2022, Prayagraj, India. DOI: https://doi.org/10.1007/978-3-031-23724-9_39

Vishnoi V.K., Kumar K., Kumar B. 2021. Plant disease detection using computational intelligence and image processing. Journal of Plant Diseases and Protection 128: 19–53. DOI: https://doi.org/10.1007/s11119-019-09703-4

Wei D., Wei X., Tang Q., Jia L., Yin X., Ji Y. 2023. RTLSeg: A novel multi-component inspection network for railway track line based on instance segmentation. Engineering Applications of Artificial Intelligence 119: 105822. DOI: https://doi.org/10.3390/s23041779