scientific reports

Check for updates

OPEN

Leveraging YOLO deep learning models to enhance plant disease identification

Yousef Alhwaiti[1]✉, Muntazir Khan[2], Muhammad Asim[2], Muhammad Hameed Siddiqi[1], Muhammad Ishaq[2] & Madallah Alruwaili[1]

Early automation in identifying plant diseases is crucial for the precise protection of crops. Plant diseases pose substantial risks to agriculture-dependent nations, often leading to notable crop losses and financial challenges, particularly in developing countries. Symptoms such as chlorosis, structural deformities, and wilting, characterize these diseases. However, early identification can be challenging due to symptoms similarity. Researchers using artificial intelligence (AI) for plant disease classification, challenges like data imbalance, symptom variability, real-time performance, and costly annotation hinder accuracy and adoption. This work introduced a novel approach using the You Only Look Once (YOLO) deep learning model, chosen for its exceptional accuracy and speed. The study focuses on analyzing YOLO models, specifically YOLOv3 and YOLOv4, to identify fruit plant diseases. This work examines healthy peach and strawberry leaves, as well as peach leaves affected by bacterial spots and strawberry leaves with scorch disease. These models underwent thorough training using data from the publicly accessible Plant Village dataset. The simulation results were highly promising, numerically YOLOv3 model achieved 97% accuracy and a Mean Average Precision (mAP) of 92%, within a total detection time of 105 s. In comparison, the YOLOv4 model outperformed, with a 98% accuracy and an impressive mean average precision of 98%, all while completing the detection process in just 29 s. YOLOv4 demonstrated lower complexity, significantly faster, and more precise performance, especially in detecting multiple items. Serving as an efficient real-time detector, it holds the potential to transform plant disease diagnosis and mitigation strategies, ultimately leading to increased agricultural productivity and enhanced financial outcomes for developing nations.

Keywords Plant diseases detection, YOLO, Deep learning, Real-time detector, Image classification, Mean average precision

Agriculture plays a crucial role in every economy, relying on the cultivation of fruits as a foundational aspect. Fruits, being a high-value crop, hold considerable sway over the economic landscape of any nation. Despite being integral to the daily lives of nearly everyone, a significant portion of fruit yields fall victim to various diseases. Many fruit plants exhibit recurring disorders, particularly in their leaves, showcasing various ailments. Identifying these disorders in their initial phases is crucial, as the effectiveness of treatment options is most pronounced during this vital period. Plant leaves and stems exhibit pivotal indicators of these diseases, frequently stemming from infections caused by bacteria, fungi, and viruses[1,2]. The consequences of plant diseases transcend the field of horticulture, encompassing plant mortality, diminished crop productivity, and significant financial losses[3,4]. Manifestations of plant diseases include various changes, encompassing structural, physiological, and color shifts, among other noticeable traits[5,6]. Precise diagnosis of these ailments is highly significant given the considerable financial losses involved and the potential decrease in agricultural output[7–9]. Consequently, the accurate identification of plant diseases continues to be a complex and critical challenge.

Bacterial, viral, and fungal infections are the leading contributors to leaf diseases. These conditions exhibit diverse visible symptoms on a plant's leaves or stems, facilitating disease diagnosis[10,11]. The repercussions of these diseases have been detrimental to crop production, making the expansion of crops a distant goal. Plant diseases manifest in various forms, making them difficult to recognize without specialized assistance, particularly for farmers in developing nations with limited access to expertise due to inadequate infrastructure and awareness. Real-time plant disease identification is challenging with traditional machine learning and

[1]College of Computer and Information Sciences, Jouf University, Sakaka, Aljouf, Kingdom of Saudi Arabia. [2]Institute of Computer Sciences and Information Technology, University of Agriculture, Peshawar, Pakistan. ✉email: ysalhwaiti@ju.edu.sa

deep learning approaches because of their limited performance and high computational demands. As a result, insufficiently addressing various plant diseases leads to significant production losses. Early disease detection is especially difficult due to overlapping symptoms on leaves and stems. Rural farmers often struggle to manage these diseases due to a lack of necessary knowledge. The agricultural community stands to gain significantly from an automated system capable of identifying diseases in their early stages[12–15]. Early diagnosis is crucial for effective treatment. While farmers traditionally relied on human expertise for manual disease detection, recent technological advancements, particularly in machine learning and artificial intelligence, now enable the identification of diseases even in their initial stages[16]. Achieving such early detection requires vigilant care and monitoring.

Historically, plant disease research has been centered in the field of plant biology. Traditionally, human experts depend on their knowledge for disease diagnosis, a manual process characterized by labor-intensive efforts and susceptible to inefficiencies and misdiagnoses. Given these challenges, the need for early disease recognition and detection has become crucial[17]. Consequently, automated approaches, providing efficient and objective analyses, have become invaluable tools for achieving high accuracy in the detection of plant diseases[18–20]. Numerous investigations have utilized deep learning methods, particularly Convolutional Neural Networks (CNNs), and machine learning models to detect a broad spectrum of plant diseases. Deep learning approaches have captured considerable interest from researchers owing to their ability to achieve outstanding accuracy across diverse applications[21–25]. Despite numerous studies that classify plant diseases using various datasets for detection, there is a noticeable scarcity of research dedicated to achieving highly precise plant disease detection. As a result, the urgency for a swift, highly efficient, real-time, and multi-feature detection technique persists.

In this study, we employ the You Only Look Once (YOLO) object detection model, renowned for its real-time efficiency and high effectiveness, to address the identification of plant diseases in fruit trees. Among the multiple variants of YOLO, our research distinguishes itself by conducting a comprehensive performance analysis of YOLOv3 and YOLOv4 models, specifically tailored for detecting diseases in fruit leaves. We rigorously evaluate the proposed models across a range of parameters, including accuracy, F-measure, precision, recall, and mean Average Precision (mAP), within the context of diverse plant leaf disease detection. Additionally, we explore the model's computation efficiency and its ability to accurately localize affected areas. Despite the existing body of research, no prior studies have presented such a robust solution for highly accurate plant disease detection. This work also suggests potential enhancements for future iterations of YOLO models to further improve their application in agricultural contexts.

Our research introduces an innovative approach to improving the detection of diseases in fruit plants using the YOLO framework. The real-time capabilities of YOLO, combined with its exceptional speed and accuracy, make it a preferred choice among researchers for various object detection tasks. Our study offers several significant contributions:

- We present an automated localization technique designed to identify affected areas within the entire image collection. Utilizing advanced deep learning models, specifically YOLOv3 and YOLOv4, we perform disease detection and classification.
- This study considers only four classes from the publicly available Plant Village dataset, which comprises 25 classes of healthy and diseased plants across 9 different plant species and contains thousands of instances.
- To evaluate the effectiveness of these models in identifying a range of plant leaf diseases, we assess their performance using various parameters, including accuracy, F-measure, precision, recall, and mean Average Precision (mAP).

The remainder of this paper is structured as follows: Section "Related Work" provides a comprehensive review of the relevant literature. Section "Methods and materials", explores the approach and techniques employed in this research. Section "Numerical Results" details our research methodology, and presents the results and their analysis. Finally, Section "Conclusions"concludes with our findings and outlines potential avenues for future research.

Related work

Deep learning models are an essential component of computer vision applications, such as tasks like object detection and image categorization, due to their ability to quickly and effectively interpret visual data[26–28]. Several researchers have previously shown that deep learning and machine learning models performed better than humans and produced more accurate findings than those obtained by human experts[29–34]. Such models utilize the power of artificial neural networks and learn complex features from image datasets. The use of deep learning techniques for the detection and classification of plant diseases is explored in this section of the literature.

A study in[4] used a deep convolution neural network method for diagnosing rice disease. The data set used in this study contains both healthy and sickly leaves, which consist of a total of 500 images. The experiment results show that the model used in this paper achieved a 95.48% accuracy rate which gives higher convergence rates and better disease detection.

The research work in[5] used RNN for disease detection and the FUZZY k-mean clustering model for the classification process in simulation to identify the banana leaf disease. They independently gathered the data, which includes the two primary banana diseases banana speckle and banana sigatoka to evaluate the effectiveness of the suggested model. Both banana leaves and fruits were used to train the suggested model to test its effectiveness and assure superior detection outcomes. To speed up the procedure, this study created a way for storing the data in the database. The total simulation results demonstrate that the proposed model was highly effective in this study. Similarly in[9], YOLO (You Only Look Once) oriented CNN model was used to detect leaf disease. YOLO was chosen because of its ability to identify multiple diseases in a single image. Furthermore[19]

proposes a Resnet-50 residual network model to identify tomato leaf disease. In this paper convolution layers were used to extract the leaf disease feature, and following repeated learning, to classify tomato leaf disease.

In a previous study[35], a convolutional neural network was employed with the Plant Village dataset. The model utilized achieved an accuracy of 85.3%. However, it's worth noting that this model had a limitation, as it required an extended training phase and could only classify single leaves on a uniform background with the leaves facing upward. Similarly, in another investigation[36], the author introduced an innovative approach for identifying plant leaf diseases using a combination of four deep learning models and a stacking classifier. The dataset included 10 different classes comprising both healthy and diseased plant leaves, totaling 36,258 images. Python 3 was used to implement their proposed CNN models, including Inception, ResNet, DenseNet, and Inception-ResNet for feature extraction, with a stacking approach employed for classification. The comprehensive simulation results revealed an impressive accuracy rate of 87% for the proposed model.

In a different study[37], image processing and genetic algorithms were utilized to identify plant leaf diseases. Genetic algorithms were employed for crop disease diagnosis and image segmentation. Various photo segmentation algorithms were applied to categorize and automatically detect diseases. The simulation results demonstrated that the proposed model achieved high performance. In[38], the author presented a CNN-based YOLO model for identifying bacterial spot disease within bell pepper leaves. About 100 pepper bell leaves were collected as a dataset from personal gardens to assess the model's performance. The simulations were conducted using TensorFlow and Keras-based Python libraries. The results from all simulations indicated an impressive 96.78% accuracy rate for detecting bacterial spot disease in bell pepper leaves.

Another study[39] proposed a CNN model with ReLU activation, combined with K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) algorithms, to identify and categorize various plant diseases. The dataset used in this study consisted of 20,000 photos across 19 classes to evaluate the model's performance. The study also incorporated a manual technique and included logistic regression for data preprocessing. The simulation results showcased that the suggested model achieved an impressive 98% accuracy rate for identifying various plant diseases. Furthermore, in[40], a novel technique based on deep learning models like LeNet was introduced for identifying tea plant leaf diseases. Table 1 provides a comprehensive summary of the existing literature about the detection of plant diseases using a diverse range of machine learning and deep learning models. The studies featured in the table encompass a variety of plant diseases, including those affecting banana leaves and fruits, rice, tomato plants, bell pepper leaves, general plant leaves, and tea leaves, and highlight the limitations of the study.

Notably, some models have exhibited remarkable accuracy rates, such as Resnet-50, which achieved an impressive accuracy of 98.0%, and Support Vector Machine (SVM) and k-Nearest Neighbors (KNN) both reaching 98% accuracy in detecting plant leaf diseases.

Furthermore, certain research endeavors have shown enhanced performance of machine learning models in the realm of plant disease detection. For instance, the utilization of LeNet has yielded promising results in the context of tea leaf disease identification, while YOLO has proven effective in enhancing accuracy across various applications. Among these models, YOLO stands out as a compelling choice for plant leaf detection, thanks to its combination of high accuracy and real-time performance. In this research, we propose the utilization of a deep learning-based YOLO model for the identification of strawberry and peach plant leaf diseases. This model has been widely acknowledged in literature for its versatility in tackling diverse object detection tasks.

Our study aims to evaluate the performance of the suggested model across multiple parameters. Specifically, we focus on identifying strawberry and peach fruit leaf diseases, recognizing the YOLOv4 model as a potent tool for effectively detecting diseases in fruits, plants, and leaves.

Methods and materials

Various techniques were used by different researchers for different plant disease detection. All these techniques have their performance in different applications. This study proposes deep learning-based YOLO-v3 and YOLO-v4 models for fruit plant leaf disease detection, given in the following sub-sections. Figure 1 presents an overview of our research methodology, which illustrates the proposed research flow diagram.

Ref	Dataset	Model	Results	Limitations
[5]	Banana leaf and fruit diseases	ANN	High accuracy	Not suited for real-time
[15]	Plant disease based on hyperspectral reflectance	SVM	97% accuracy	Custom dataset, Low accuracy on multiple classes
[19]	Tomato plant diseases	Resnet-50	98.0% accuracy	High Train/Test ratio – 90/10
[20]	Plant disease	YOLO		Numerical results not available
[35]	Plant village dataset	CNN	85.3% accuracy	Less Accuracy
[36]	Plant village dataset	Inception, ResNet, DenNet	87% accuracy	High Complexity, not suitable for real-time
[37]	Plant leaf disease	Genetic algorithm	Less Performance	Not suited for real-time analysis
[38]	LabelStoma: A for stomata detection	YOLOv3	High Accuracy	High Test/Train ratio – 85/15
[39]	Bell pepper plant	YOLOv5	High performance	Missing numerical results

Table 1. Summary of the related work.

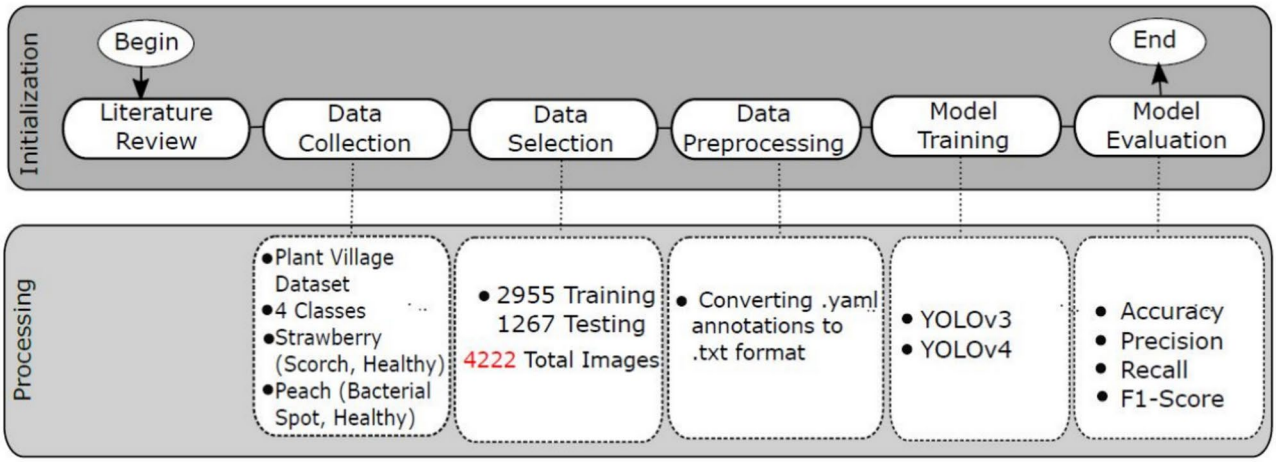

Fig. 1. Research Methodology for Disease Detection in Fruit Plants Using YOLO Models.

S. No	Fruit Type	Type	Number of Images
1	Peach	Healthy	360
		Bacterial Spot	2297
2	Strawberry	Healthy	456
		Leaf Scorch	1109
Total Number of Images			4222

Table 2. Class-wise image distribution of the data used for training and testing of models.

Data collection

We employed the Plant Village dataset[41] to train our models, focusing on four distinct classes related to two types of fruit leaves: healthy and bacterial spot-affected peach leaves and healthy and scorched strawberry leaves. Our dataset comprises a total of 4,222 images used for both training and model evaluation. Specifically, we have 2,657 Peach leaf images and 1,565 Strawberry leaf images. Further details about healthy and infected images can be found in Table 2.

Classes

In this paper, we focus our attention on a subset of just four specific classes from the Plant Village dataset[42]. However, it is important to highlight that this dataset encompasses a broader spectrum of 37 diverse categories, encompassing various fruit and vegetable leaf types. This extensive collection serves as a valuable resource for exploring the automation of disease detection and classification across this wide range of plant species. The selected classes from the dataset are briefly discussed below:

Peach bacterial spot

Peaches, nectarines, apricots, and plums are susceptible to a severe bacterial infection known as Peach Bacterial Spot, caused by the bacterium Xanthomonascampestris PV. Pruni. This illness manifests in various ways, including fruit spots, leaf blight, and twig cankers. Additionally, affected fruit may exhibit symptoms such as pitting, cracking, gumminess, and water-soaked tissue, which can make them more vulnerable to brown rot, rhizopus, and other fungal infections. Severe leaf spot infestations can lead to early defoliation, while extreme defoliation can result in fruit splitting, shrinking, or sunburn. Furthermore, early leaf loss can weaken the trees and reduce their winter hardiness. Peach Bacterial Spot images are presented in Figure 2 for ready reference.

Peach healthy.

Prunus persica, a deciduous peach tree, has its origins in the eastern Chinese province of Zhejiang, where it was first domesticated and cultivated. This remarkable tree yields a variety of delicious fruits, most referred to as peaches, with some being known as nectarines. The specific name 'persica' alludes to its widespread cultivation in Iran, from where it was introduced to Europe. Belonging to the Rosaceae family, it is a member of the Prunus genus, which also includes cherries, nuts, apricots, and plums. Figure 3 showcases images of healthy peach specimens.

Strawberry leaf scorch

Strawberry Leaf scorch is a disease that affects strawberry plants, particularly those of the Fragaria x ananassa species, which is commonly cultivated for its sweet and juicy berries. This disease is caused by a bacterium

Fig. 2. Peach bacterial spot leaves.

Fig. 3. Peach healthy leaves.

Fig. 4. Strawberry Scorch leaves.

known as Xanthonmous fragaries. Common symptoms used to identify this disease include reddish lesions, marginal necrosis, leaf curling, and reduced fruit production. Strawberry growers need to monitor their plants regularly for signs of disease and take prompt action if Strawberry Leaf Scorch is detected. Early intervention and preventive measures are essential for managing this disease and protecting strawberry crops. Figure 4 illustrates the leaf scorch.

Strawberry healthy

The genus Fragaria, a member of the Rosaceae family, encompasses over 100 diverse flowering plant species, many of which bear edible fruits. Cultivated strawberries are extensively grown worldwide due to their popularity. Strawberries naturally thrive in temperate regions of the Northern Hemisphere. In Figure 5, you can observe healthy strawberry leaves for reference.

Processing and data selection

This study utilizes data from the publicly available Plant Village dataset[41], accessible on Kaggle. To ensure compatibility with our model, all images are resized to a uniform 416x416 size before training and testing. The following preprocessing steps are employed to augment the data, as outlined below:

- We leverage the powerful deep learning library, Keras, which offers an Image Data Generator facilitating image augmentation through standard techniques like rescaling, shearing, zooming, and horizontal flipping.

Fig. 5. Strawberry healthy leaves.

- To maintain consistency across all images, a rescaling factor of 1/255 is applied, thereby normalizing the pixel range of all data images to a range of 0 to 1.
- The total of 4222 augmented images is divided into training and testing datasets.
- The YOLO environment is established by cloning the corresponding YOLO variant's repository from GitHub, followed by setting up the necessary data and directory structure.
- Finally, YAML configuration files are prepared. These files include essential information such as training and testing file paths and the number of classes required for training the models. This comprehensive preprocessing ensures that the data is well-prepared and suitable for model training and evaluation.

Model training

This study considers the two variants of the YOLO framework, i.e., YOLOv3 and YOLOv4 are proposed in this research work for plant disease detection.

YOLO-v3

YOLOv3, an evolution within the YOLO (You Only Look Once) family, made its debut in 2018[43]. It stands as a profound deep neural network model designed for swift and precise object detection in real-time within images or video streams. YOLOv3 extends upon its predecessor by incorporating a more substantial backbone network, a feature pyramid network, and multi-scale prediction, enabling it to adeptly detect objects of diverse sizes. It also integrates techniques such as batch normalization, shortcut connections, and spatial attention to further elevate accuracy and resilience. This versatile framework has garnered widespread adoption in computer vision applications spanning surveillance, autonomous driving, robotics, and even novel use cases like post-natural disaster building damage assessment.

The primary architecture of YOLOv3 is compartmentalized into three core segments: The backbone network, the detection head, and the post-processing module. The backbone network, initially tasked with feature extraction from input images, leverages a modified DarkNet-53 architecture. This backbone employs a sequence of convolutional layers intercepted with batch normalization and leaky ReLU activations to enhance learning and prevent overfitting. Max-pooling layers are strategically integrated to progressively reduce the spatial dimensions of the feature maps, thereby concentrating the salient feature.

For estimating bounding boxes and class probabilities within images, the detection head comes into play. It processes the multi-scale feature maps generated by the backbone network through a series of convolutional layers. This multi-scale approach allows the model to detect objects of varying sizes more effectively. Each detection layer is responsible for predicting a fixed number of bounding boxes, which is typically determined by the number of anchor boxes.

The post-processing module plays a pivotal role in refining the detections. It filters detections based on their confidence scores and employs non-maximum suppression (NMS) to eliminate overlapping and redundant results. Detections with low confidence scores are discarded, ensuring the output is both accurate and concise. The NMS algorithm ensures that only the most relevant bounding boxes are retained, which reduces the likelihood of duplicate detections. Additionally, the integration of residual blocks in the backbone architecture allows for improved gradient flow during training, facilitating deeper networks without suffering from the vanishing gradient problem. The flow diagram of YOLOv3, coupled with residual blocks and the detection module is presented in Figure 6. This combination of advanced post-processing makes YOLOv3 a powerful model for real-time object detection.

YOLO-v4

YOLOv4, an upgraded iteration of the YOLOv3 object detection model, stands as a remarkable achievement in the field. Introduced in 2020 and credited to[44], this model is renowned for its exceptional precision and speed. YOLOv4's architecture comprises three key components: the backbone network, the neck, and the head, each with distinctive traits and design considerations.

YOLOv4's backbone network builds upon the foundation of CSPDarknet53, a variation of the DarkNet architecture utilized in YOLOv3. Notably, CSPDarknet53 incorporates cross-stage partial connections, enhancing feature reuse while minimizing network parameters. This network's primary role is to extract features from input images and reduce spatial resolution, achieved through a series of convolutional layers and downsampling techniques.

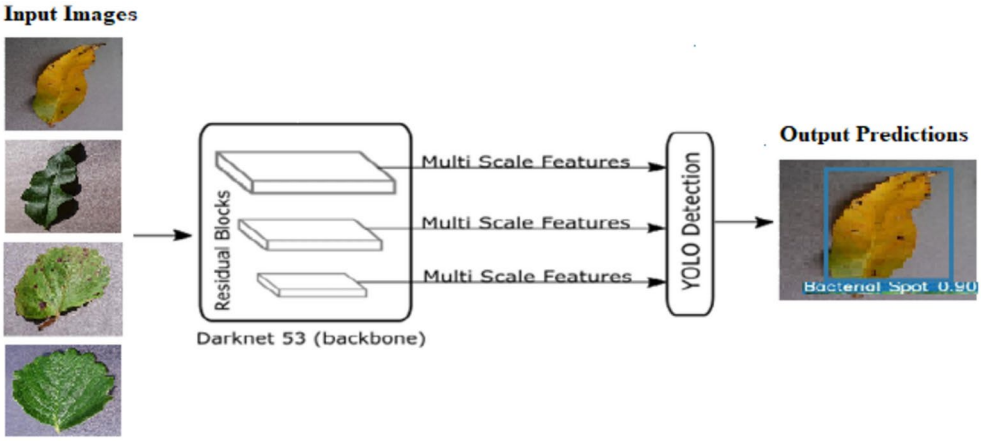

Fig. 6. Block diagram of YOLOv3 model for Plant Disease Detection.

YOLOv4's neck incorporates spatial attention modules, strategically designed to focus on essential image areas while suppressing extraneous details. These spatial attention modules are introduced after several layers of the backbone network, enhancing feature maps through channel and spatial attention mechanisms.

The head in the YOLOv4 model is responsible for predicting bounding boxes and class probabilities for objects within images, the head of YOLOv4 consists of multiple detection layers. Each layer predicts a specific number of bounding boxes, often based on the predetermined number of anchors for that layer. Additionally, the head integrates several convolutional layers to analyze feature maps generated by the neck and backbone networks. The final output of the model comprises bounding boxes and associated class probabilities for detected objects.

Beyond these core components, YOLOv4 incorporates advanced training techniques. These include cut mix and mosaic data augmentation, which bolster model accuracy and robustness. Additionally, YOLOv4 adopts a modified variant of the non-maximum suppression algorithm known as weighted box fusion (WBF).

To recognize objects in real-time, the model employs a deep neural network that has been trained on a sizable dataset of images. The input image is divided into a grid by YOLOv4 and numerous bounding boxes, each with a different confidence score but instead of class prediction, are predicted for each grid cell. The output is a collection of bounding boxes, each with a confidence score and class label. The capability of YOLOv4 to find small objects is another crucial trait. Spatial Pyramid Pooling (SPP), a method that enables the model to capture features at various scales and resolutions, is used to accomplish this. As a result, YOLOv4 is especially advantageous for use in robotics, self-driving cars, and surveillance.

Bounding Box Prediction: YOLOv4 employs a grid-based approach to dissect the input image, resulting in multiple bounding boxes per cell. For every such bounding box, the model predicts both its confidence score and class identification. Specifically, YOLOv4 furnishes predictions regarding the position, dimensions, and class label of the object encapsulated within each bounding box. This computation of bounding box 'b' in YOLO follows the formulation delineated in Eq. (1).

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned} \tag{1}$$

where b_x and b_y **represents** bounding box center coordinates, b_w and b_h represent width and height. The width and height of each bounding box are predicted as offsets from predefined cluster centroids (anchor boxes), as depicted in Fig. 7. The center coordinates of the bounding box are determined relative to the grid cell where the object is detected, using a sigmoid function to ensure the coordinates fall within the bounds of the grid cell. This approach allows YOLO to predict precise object locations within the image.

Non-Maximum Suppression: YOLOv4 employs the non-maximum suppression (NMS) method to prevent duplicate detections. Bounding boxes that significantly overlap with other bounding boxes with greater confidence scores are removed by NMS. A final set of bounding boxes is the result, and these bounding boxes are very apt to contain interesting objects as shown in Eq. (2).

$$\mu_{t,k}^{-l}(\varnothing_t) = \frac{1}{k}\sum_{T}^{k-1} \mu_{t-T}^{l}(\varnothing_t) \tag{2}$$

Output: Each bounding box in the output of YOLOv4 has a class label and a confidence value. To draw attention to the objects that the model has identified, these bounding boxes can be drawn on the input picture. The step-by-step procedure of the employed model used for this research has been presented in Algorithm 1.

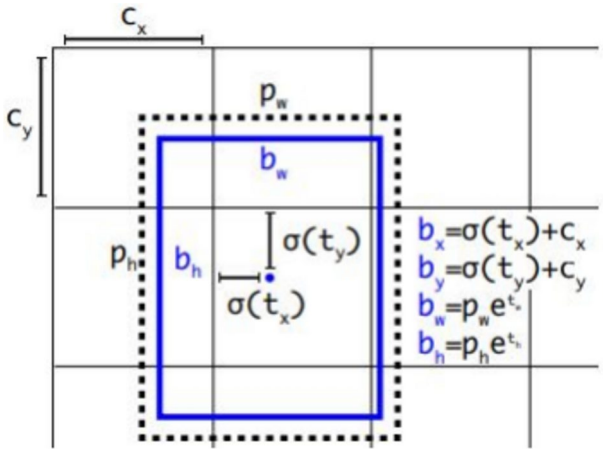

Figure. 7. Bounding boxes with dimension priors and location predictions.

Input: Plant Village Dataset

Output: Disease Classification - Strawberry (Leaf scorch, Healthy), Peach (Bacterial Spot, Healthy)

Step 1: *Begin*

Step 2: *Initialize Darknet*

Step 3: *Input data to the proposed model*

Step 4: *Assign desired weight file, image size and confidence value*

Step 5: *Start training*:

while (*current_iteration* <= *maximum_iteration*):

Train the proposed model

current_image = *readImage*()

for i from 0 to NoOfCells:

for j from 0 to NoOfCells:

current_cell = *current_image*($i: i + step, j: j + step$)

Generate new weights file

$$current_{iteration} = current_{iteration} + 1$$

End while

Step 6: *Start training from the previous saved state*:

if (*current_iteration* >= *maximum_iteration*):

prediction_class_array$[i, j]$ = *class_predictor*(*current_cell*)

predictions_bounding_box_array$[i, j]$ = *bounding_box_predictor*(*current_cell*)

$prediction = [prediction\_bounding\_box\_array[i, j, best_{bounding_{box}}, [0:4], predicted_{class}]$

else: *Go back to Step 5 with new assigned weight*

Step 7: *Test data*

final_predictions.append(*prediction*)

print final_predictions

Step 8: *End*

Algorithm 1. Proposed YOLO model Pseudo code for detecting leaf disease.

Experimental results

In this study, we conducted our experiments on a Windows 10 computer equipped with an Intel Core i7 processor running at 2.0 GHz and 16 GB of RAM. The dataset was employed for training and testing the machine learning algorithms, and this was achieved using the Keras Python library. Model training was performed on Google Colab Pro, leveraging the capabilities of a GPU. Specifically, we employed two distinct models, YOLOv4 and YOLOv3, for the detection of plant leaf diseases in fruit plants, assessing their performance in terms of accuracy, *F*-measure, recall, and precision.

In this study, we employ a range of performance parameters to assess the effectiveness of the proposed model. Below, we provide a concise overview of the performance parameters used for evaluating the models.

Confusion matrix
In this study, we employ the confusion matrix as a representation to analyze the alignment of categorizations under specific constraints. The dimensions of the matrix, denoted as M = n*n, correspond to the number of classes being considered. As such, the summation of all positively identified values, encompassing both true positives (TP) and false positives (FP), is considered the most precise measure of identification accuracy. True positives indicate instances where an element is correctly associated with a class it genuinely belongs to, while false positives signify cases where an element is inaccurately linked to a specific class. Any instances not falling into these categories, which include erroneous false negatives (FN) and false positives (FP), are categorized as 'rejected.' For a visual representation of the confusion matrix format, refer to Table 3.

Accuracy
The accuracy of our model determines its effectiveness. It demonstrates how secure and portable our methodology is when detecting both negative and positive values and how close to the true values our model has been returning the results. The accuracy of the proposed model is calculated using Eq. (3).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Precision
Count the instances of positive values that have been accurately retrieved, signifying the total number of values authentically associated with a particular class that were successfully identified. Precision is then determined by dividing the count of true positives by the total of all correctly detected values and classes. This precision calculation is represented as follows in Eq. (4).

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Recall
The True Positive Rate (TPR) is commonly referred to as recall and is calculated by dividing the total number of true positive values by the sum of the true positive and false negative values. Recall is a quantitative indicator that demonstrates the thoroughness of the outcome. The number of accurately detected is known as the true positive rate. Good recall indicates that the data is efficiently and correctly recovered. The recall is calculated using Eq. (5).

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

F-measure
If we compare two models, it is quite difficult to compare a model with good precision and high recall to one with poor precision and high recall. Hence, we employ the *F*-score or *F*-measure to accomplish this. Instead of using arithmetic means, the F-measure uses harmonic means and is calculated using Eq. (6).

$$F - Measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{6}$$

Mean average precision
The metric mAP@0.5 represents the mean Average Precision (mAP) at an Intersection over a Union (IoU) threshold of 0.5, while mAP@0.5:0.95 stands for the average mAP calculated across various IoU thresholds ranging from 0.5 to 0.95. The calculation is provided in Eq. (7).

$$mAP = \frac{1}{n} \sum_{n=1}^{k=n} AP_k \tag{7}$$

Numerical results

Performance evaluation of YOLOv3 model

The performance evaluation of the YOLOv3 model is meticulously presented, with individual assessments for each class featured in Figure 8 and elaborated upon in Table 4. Simultaneously, Figure 9 (a)-(d) offers a bounding box prediction of the YOLOv3 model. Based on the insights garnered from Figure 8 and Table 4, aligned with

		Predicted class	
		Related (P)	**Non-related (N)**
Actual class	Related (P)	True positive (TP)	False negative (FN)
	Non-related (N)	False positive (FP)	True negative (TN)

Table 3. Confusion matrix.

```
 calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
4530
 detections_count = 4295, unique_truth_count = 985
class_id = 0, name = Strawberry___healthy, ap = 92.39%          (TP = 986, FP = 24)
class_id = 1, name = Strawberry___Leaf_scorch, ap = 92.68%      (TP = 647, FP = 19)
class_id = 2, name = Peach___healthy, ap = 92.92%               (TP = 872, FP = 18)
class_id = 3, name = Peach___Bacterial_spot, ap = 92.73%        (TP = 802, FP = 15)

 for conf_thresh = 0.25, precision = 0.97, recall = 0.97, F1-score = 0.97
 for conf_thresh = 0.25, TP = 3307, FP = 76, FN = 69, average IoU = 89.22 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.465674, or 46.56 %
 Total Detection Time: 105 Seconds
```

Fig. 8. Numerical results in terms of a confusion matrix for YOLOv3.

Classes	FP	TP	AP
Strawberry_healthy	24	986	92.39
Strawberry_leaf_scorch	19	647	92.68
Peach_healthy	18	872	92.92
Peach_bacterial_spot	15	882	92.73

Table 4. Class-wise Performance Evaluation of YOLOv3.

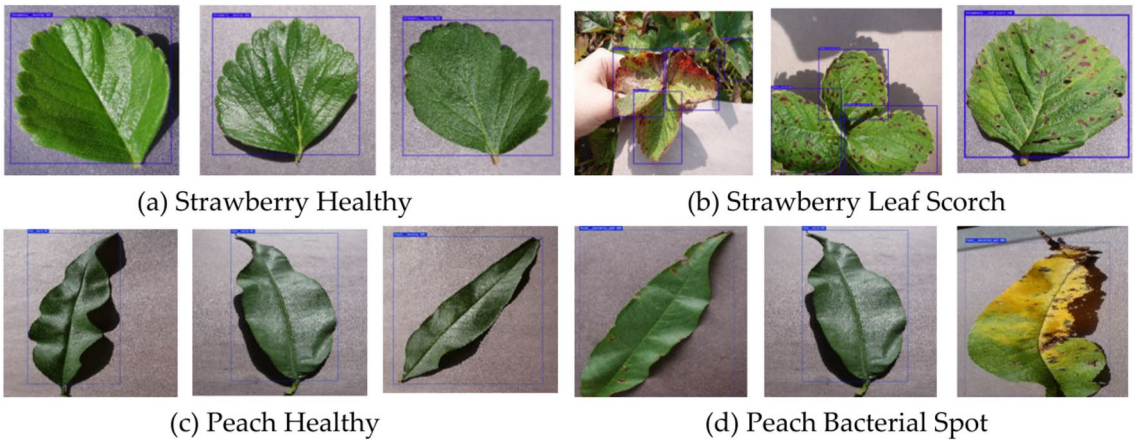

Fig. 9. Bounding Box prediction of Strawberry and Peach diseases detection using YOLOv3 model.

the previously discussed performance metrics, the YOLOv3 model delivers remarkable statistics, boasting a precision rate, accuracy, recall, and *F*-measure all resting at a commendable 97%, while displaying a negligible loss of 0.075.

Table 4 takes a granular approach, offering an exhaustive class-wise evaluation of the YOLOv3 model's performance. To illustrate, consider the 'strawberry healthy' class, which achieves an impressive average precision of 92.3%. This class is backed by 986 True Positive (TP) detections, with only 24 instances of False Positives (FP). Similarly, the 'strawberry leaf scorch' class attains an average precision of 92.6%, driven by 647 TP detections and a mere 19 FP detections. The 'Peach healthy' class shows an average precision of 92.92%, substantiated by 872 TP detections and just 18 FP detections. Lastly, the 'Peach Bacterial Spot' class secures an average precision of 92.73%, supported by 802 TP detections and a mere 15 FP detections.

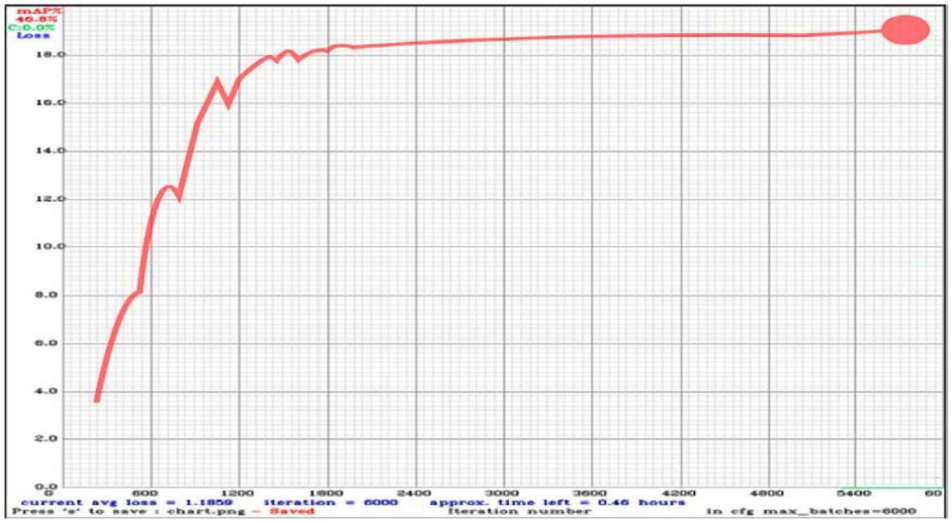

Fig. 10. The number of iterations required to achieve accuracy for the YOLOv3 Model.

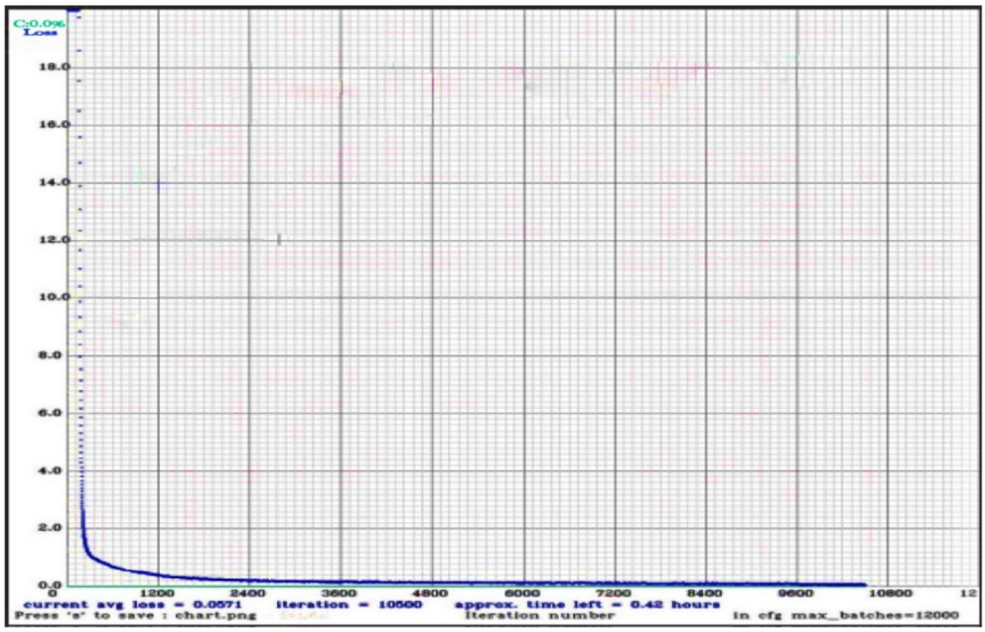

Fig. 11. Loss curve in terms of number of iterations for the YOLOv3 model.

Figure 10 and Figure 11 depict the iteration count needed to attain specific accuracy or loss thresholds for the YOLOv3 model. The YOLOv3 model rapidly ascends to its peak accuracy with only a handful of iterations and then exhibits stable convergence for subsequent batches. A similar intuition can be drawn for the loss function as shown in Figure 10.

Performance evaluation of YOLOv4 model

The performance evaluation of the YOLOv4 model, as illustrated in Figure 12 and detailed in Table 5, provides a clear depiction of the true and false positives generated during the testing phase. In the subsequent sections, we delve into a comprehensive discussion and summary of these results.

Moreover, Figure 12 provides a comprehensive performance evaluation of YOLOv4, breaking down the assessment for each class. As is clear from Table 6, the employed model consistently achieves a precision rate of 98% across all categories. When considering a broader range of performance metrics, including standard parameters such as accuracy, recall, precision, F_1-score, and mean average precision (mAP), the results for the proposed model are as follows: a precision rate of 98%, an accuracy of 98%, a recall of 99%, an F-measure of 99%, and a loss rate of 0.00534.

```
 calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
4264
 detections_count = 4295, unique_truth_count = 985
class_id = 0, name = Strawberry___healthy, ap = 98.89%           (TP = 980, FP = 9)
class_id = 1, name = Strawberry___Leaf_scorch, ap = 98.73%       (TP = 640, FP = 11)
class_id = 2, name = Peach___healthy, ap = 98.93%                (TP = 865, FP = 7)
class_id = 3, name = Peach___Bacterial_spot, ap = 98.88%         (TP = 795, FP = 10)

 for conf_thresh = 0.25, precision = 0.98, recall = 0.99, F1-score = 0.99
 for conf_thresh = 0.25, TP = 3280, FP = 37, FN = 28, average IoU = 89.22 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.495674, or 49.56 %
 Total Detection Time: 29 Seconds
```

Fig. 12. Numerical results in terms of a confusion matrix for the YOLOv4.

Classes	FP	TP	AP
Strawberry_healthy	09	980	98.89
Strawberry_leaf_scorch	11	640	98.73
Peach_healthy	07	865	98.93
Peach_bacterial_spot	10	795	98.88

Table 5. Performance Evaluation of the different classes for YOLOv4.

References	Accuracy	F-Measure	Precision	Recall	mAP
[45]	x	x	86.5	x	YOLOv5 = 70
[46]	DBA_SSD = 91 Faster RCNN = 89 YOLOv4 = 92	x	x	x	DBA_SSD = 92.20 Faster RCNN = 90.10 YOLOv4 = 87.57
[47]	Resenet-50 = 97.84	x	x	x	YOLOv5 = 95.17
Optimized YOLOv5[48]	x	92.97	93.73	92.94	x
YOLOv3 (Our)	97	96	97	97	92
YOLOv4 (Our)	98	99	98	99	98

Table 6. Results comparison of YOLOv3 and YOLOv4 basis on various parameters with different studies.

Likewise, Table 5 presents a class-wise evaluation of the YOLOv4 model's performance. For instance, the 'strawberry healthy' class achieves an average precision of 98.89%, with True Positive (TP) values of 980 and False Positive (FP) values of 9. Similarly, the 'strawberry leaf scorch' class achieves an average precision of 98.73%, with TP values of 640 and FP values of 11. The 'Peach healthy' class attains an average precision of 98.93%, with TP values of 865 and FP values of 7. Additionally, the 'Peach Bacterial Spot' class demonstrates an average precision of 98.88%, with TP values of 797 and FP values of 10. To further illustrate our findings, YOLOv4 has a low failure rate due to its high learning rate. Figure 13(a), (b), (c), and (d) displays the visual results of image recognition conducted by the proposed YOLOv4 model, encompassing categories such as 'strawberry healthy,' 'strawberry leaf scorch,' 'peach bacterial spot,' and 'peach healthy.'

Figure 14 and Fig. 15 depict the iteration count needed to attain specific accuracy or loss thresholds for the YOLOv4 model. The YOLOv4 model rapidly ascends to its peak accuracy with only a handful of iterations and then exhibits stable convergence for subsequent batches. A similar intuition can be drawn for the loss function as shown in Fig. 14, as the loss is incredibly low. The loss rate of the model reduces as the accuracy of the model increases and vice versa.

Comparison of YOLOv3 and YOLOv4 model

The results of this research highlight the exceptional performance of both models, characterized by remarkably high accuracy and impressively low loss rates. Specifically, YOLOv3 attains an impressive 97% accuracy with a loss of 0.075, while the suggested YOLOv4 further elevates the bar with a striking 98% accuracy and a mere

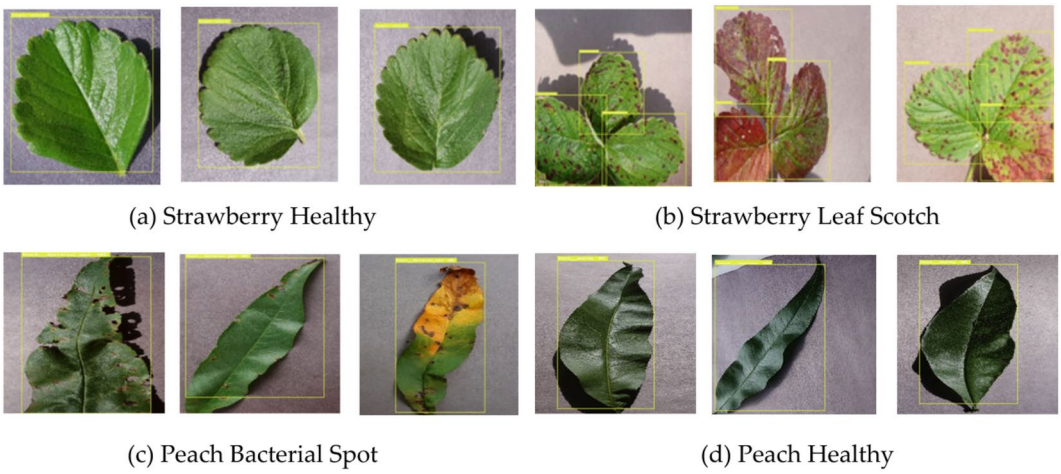

Fig. 13. Bounding box Prediction of strawberry and peach leaf detection using YOLOv4.

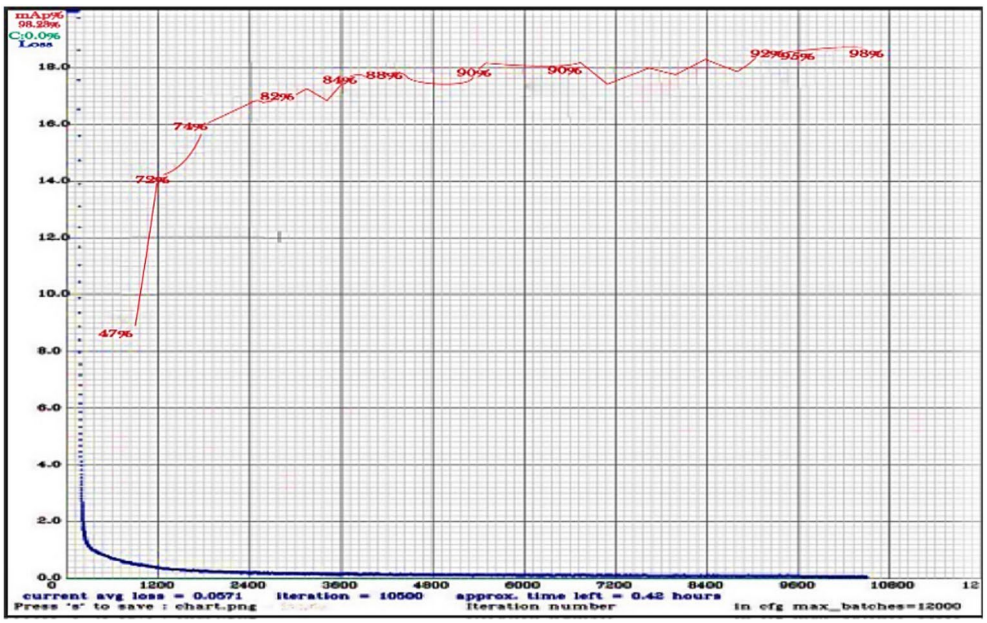

Fig. 14. The number of iterations required to achieve accuracy for the YOLOv4 model.

0.00534 loss. This stark contrast underscores YOLOv4's supremacy, showcasing not only a higher precision rate but also superior accuracy coupled with minimal loss.

Moreover, YOLOv4 excels in recall, boasting an impressive 99% rate, and delivers an *F*-Measure of 98%. In comparison, YOLOv3 records a respectable 97% recall and a 96% *F*-Measure. These metrics affirm YOLOv4's dominance across various parameters, reinforcing its superior performance. Crucially, YOLOv4's mean Average Precision (mAP) shines at an astounding 98%, while YOLOv3 lags at 92%. This substantial difference underscores YOLOv4's significant superiority in precision.

In addition, when it comes to overall detection time, YOLOv4 proves its mettle, requiring a mere 29 seconds, compared to YOLOv3's 105 seconds.

Table 6presents a comparative analysis of various object detection models based on Accuracy, F-Measure, Precision, Recall, and mAP. Reference[45]reports a precision of 86.5% and a mAP of 70% for YOLOv5. In[46], DBA_SSD, Faster R-CNN, and YOLOv4 achieve accuracy of 91%, 89%, and 92%, respectively, with corresponding mAP values of 92.20, 90.10, and 87.57. Similarly[47], highlights ResNet-50 with 97.84% accuracy and YOLOv5 with a mAP of 95.17%. Optimized YOLOv5[48] demonstrates an accuracy of 92.97%, along with precision and recall values of 93.73% and 92.94%, respectively. In comparison, our proposed models, YOLOv3 and YOLOv4, outperform the existing approaches. YOLOv3 achieves 97% accuracy with an F-Measure of 96%, Precision and Recall of 97%, and a mAP of 92%. Meanwhile, YOLOv4 further improves performance, achieving 98% across Accuracy, F-Measure, and Recall, along with 96% Precision and an outstanding mAP of 98%. As all these

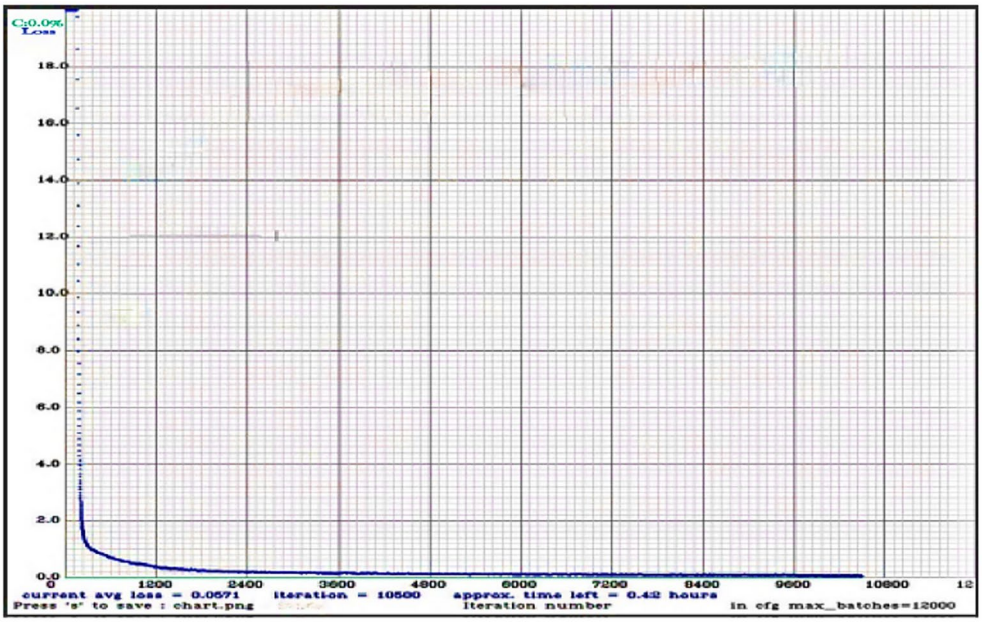

Fig. 15. Loss curve in terms number of iterations for the YOLOv4 model.

studies use the same dataset but with varying numbers of classes, the results provide a fair comparison of model performance. These findings underscore the superiority of our YOLO-based models, particularly YOLOv4, which outperforms all other methods across key metrics. The high accuracy, recall, and mAP values reinforce YOLOv4's effectiveness, making it a robust choice for real-world object detection applications.

Our research will focus on expanding the applications of the YOLOv4 model applications. We aim to leverage its capabilities for the comprehensive detection and classification of various diseases across multiple fruit leaf types, for example, Tomato leaf detection and classification[49]. As part of our innovative efforts, we plan to develop a user-friendly Android/iOS mobile application. This app will serve as a valuable tool, empowering farmers with swift and accurate disease identification, thereby enhancing their agricultural practices.

Furthermore, our research has a vision that extends to the realm of hardware integration using deep learning approaches that can provide better accuracy with lower computational complexity[42,50]. Our collective pursuit is anchored in the mission to advance disease control strategies and contribute to the sustainability of agriculture.

Conclusions

Agriculture remains a cornerstone for developing nations, yet it faces growing threats from plant diseases that can severely impact crop health and yield. The early detection of these diseases is crucial but traditional methods often lack precision and scalability. To address these challenges, this study introduces the application of deep learning models, specifically YOLOv3 and YOLOv4, for accurate and efficient detection of diseases in fruit plants. This work provides a more rigorous performance evaluation of each class, ensuring a fair comparison by utilizing the same dataset. Our research demonstrates that these models, particularly YOLOv4, significantly enhance detection accuracy and mean average precision, outperforming previous methodologies. YOLOv4 achieved an impressive 98% accuracy with a minimal loss rate of 0.00534, compared to YOLOv3's 97% accuracy and 0.075 loss rate. Additionally, YOLOv4 excels in mean average precision (mAP), achieving 98% versus YOLOv3's 92%, and operates with remarkable efficiency, reducing detection time to 29 seconds from YOLOv3's 105 seconds.

These results highlight the potential of deep learning in revolutionizing agricultural disease detection, offering faster, more precise, and scalable solutions. Future work will focus on expanding the assessment of the YOLOv4 model and newer YOLO variants to multiple classes, automating disease detection, and classifying various fruit leaves and disease localization. Additionally, we plan to integrate these models into an embedded platform to further advance agricultural practices and disease control in real-time models.

Ethical and informed consent

The datasets used have been publicly approved by physicians and experts and are freely accessible to the research community without requiring consent.

Data availability

The dataset employed in this study will be provided upon reasonable request to the corresponding author.

Received: 3 March 2024; Accepted: 25 February 2025
Published online: 07 March 2025

References

1. Sharma, P., Hans, P. & Gupta, S. C. Classification of plant leaf diseases using machine learning and image preprocessing techniques. In *2020 10th international conference on cloud computing, data science & engineering* 480–484 (Uttar, Pradesh, India, 29–31 January 2020).
2. Savary, S., Ficke, A., Aubertot, J. N. & Hollier, C. Crop losses due to diseases and their implications for global food production losses and food security. *Food secur.* **4**, 519–537 (2012).
3. Ristaino, J. B. et al. The persistent threat of emerging plant disease pandemics to global food security. *Proc. Natl. Acad. Sci.* **118**, (2021).
4. Fones, H. N. et al. Threats to global food security from emerging fungal and oomycete crop pathogens. *Nat. Food* **1**, 332–342 (2020).
5. Saranya, N., Pavithra, L., Kanthimathi, N., Ragavi, B. & Sandhiyadevi, P. Detection of banana leaf and fruit diseases using neural networks. In *Proc.2nd International Conference on Inventive Research in Computing Applications* 493–499 (Coimbatore, India, 15–17 July 2020).
6. Hossain, E., Hossain, M.F. & Rahaman, M.A. A color and texture-based approach for the detection and classification of plant leaf disease using KNN classifier. In *Proc. International Conference on Electrical, Computer and Communication Engineering*, Cox's Bazar, 1–6 (Bangladesh, 7–9 February 2019).
7. Barbedo, J. G. A. A review on the main challenges in automatic plant disease identification based on visible range images. *Biosyst. Eng.* **144**, 52–60 (2016).
8. Abu-Naser, S. S., Kashkash, K. A. & Fayyad, M. Developing an expert system for plant disease diagnosis. *Scialert* **3**(269), 276 (2010).
9. Riley, M. B., Williamson, M. R. & Maloy, O. Plant disease diagnosis. *plant health instr.* https://doi.org/10.1094/PHI-I-2002-1021-01 (2002).
10. Francis, J. & Anoop, B.K. Identification of leaf diseases in pepper plants using soft computing techniques. In *Proc.International Conference on Emerging Devices and Smart Systems*, 168–173 (Namakkal, India, 4–5 March, 2016).
11. Singh, V. & Misra, A. K. Detection of plant leaf diseases using image segmentation and soft computing techniques. *Inf. Process. Agric.* **4**, 41–49 (2017).
12. Rahman, M.A., Islam, M.M., Mahdee, G.S. & Kabir, M.W.U. Improved segmentation approach for plant disease detection. In *Proc. 1stInternational Conference on Advances in Science, Engineering and Robotics Technology*, 1–5 (Dhaka, Bangladesh, 3–5 May, 2019).
13. Schaad, N. W. & Frederick, R. D. Real-time PCR and its application for rapid plant disease diagnostics. *Can. J. plant pathol.* **24**, 250–258 (2002).
14. Thomas, S. et al. Benefits of hyperspectral imaging for plant disease detection and plant protection: A technical perspective. *J. Plant Dis. Prot.* **125**, 5–20 (2018).
15. Rumpf, T. et al. Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. *Comput. Electron. Agric.* **74**, 91–99 (2010).
16. Militante, S.V., Gerardo, B.D. & Dionisio, N.V. Plant leaf detection and disease recognition using deep learning. In *Proc. Eurasia Conference on IOT, Communication and Engineering* 579–582 (Yunlin, Taiwan, 3–6 October, 2019).
17. Sardogan, M., Tuncer, A. & Ozen, Y. Plant leaf disease detection and classification based on CNN with LVQ algorithm. In *Proc. 3rd international conference on computer science and engineering*. 382–385 (Medan, Indonesia, 20–23 September, 2018).
18. Aasha Nandhini, S., Hemalatha, R., Radha, S. & Indumathi, K. Web enabled plant disease detection system for agricultural applications using WMSN. *Wireless Pers. Commun.* **102**, 725–740 (2018).
19. Jiang, D., Li, F., Yang, Y. & Yu, S. A tomato leaf diseases classification method based on deep learning. In *Proc.Chinese Control and Decision Conference* 1446–1450 (Hefei, China, 24 August, 2020).
20. Morbekar, A., Parihar, A. & Jadhav, R. Crop disease detection using YOLO. In *2020 International Conference for Emerging Technology* 1–5 (Belgaum, India, 5–7 Jun, 2020).
21. Khan, A. et al. Optimizing connection weights of functional link neural network using APSO algorithm for medical data classification. *J. King Saud Univ. – Comput. Inf. Sci.* **34**, 2551–2561 (2002).
22. Imran, M., Khan, S., Hlavacs, H., Khan, F. A. & Anwar, S. Intrusion detection in networks using cuckoo search optimization. *Soft Comput.* **26**, 10651–10663 (2022).
23. Khan, A. et al. An alternative approach to neural network training based on hybrid bio meta-heuristic algorithm. *J. Ambient. Intell. Humaniz. Comput.* **10**, 3821–3830 (2019).
24. Yadav, A. & Vishwakarma, D. K. Sentiment analysis using deep learning architectures: a review. *Artif. Intell. Rev.* **53**, 4335–4385 (2020).
25. Otter, D. W., Medina, J. R. & Kalita, J. K. A survey of the usages of deep learning for natural language processing. *IEEE Trans. Neural Netw. Learn. Syst.* **32**, 604–624 (2020).
26. Zhao, Z. Q., Zheng, P., Xu, S. T. & Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **30**, 3212–3232 (2019).
27. Mohsen, H., El-Dahshan, E. S. A., El-Horbaty, E. S. M. & Salem, A. B. M. Classification using deep learning neural networks for brain tumors. *Futur. Comput. Inf. J.* **3**, 68–71 (2018).
28. Ashqar, B. A., Abu-Nasser, B. S. & Abu-Naser, S. S. Plant seedlings classification using deep learning. *Int. J. Acad. Inf. Syst. Res.* **3**, 7–14 (2019).
29. Mohanty, S. P., Hughes, D. P. & Salathé, M. Using deep learning for image-based plant disease detection. *Front. plant sci.* **7**, 1419 (2016).
30. Chohan, M., Khan, A., Chohan, R., Katpar, S. H. & Mahar, M. S. Plant disease detection using deep learning. *Int. J. Recent. Technol. Eng.* **9**, 909–914 (2020).
31. Ramesh, S., Hebbar, R., Niveditha, M., Pooja, R., Shashank, N. & Vinod, P. V. Plant disease detection using machine learning. In *2018 International conference on design innovations for 3Cs compute communicate control* 41–45 (Bangalore, India, 25–28 April 2018).
32. Ferentinos, K. P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **145**, 311–318 (2018).
33. Barbedo, J. G. A. Plant disease identification from individual lesions and spots using deep learning. *Biosyst. Eng.* **180**, 96–107 (2019).
34. Saleem, M. H., Potgieter, J. & Arif, K. M. Plant disease detection and classification by deep learning. *Plants* **8**, 468 (2019).
35. Tian, Y., Yang, G., Wang, Z., Li, E. & Liang, Z. Detection of apple lesions in orchards based on deep learning methods of cyclegan and yolov3-dense. *J. Sens.* https://doi.org/10.1155/2019/7630926 (2019).
36. Guan, X. A novel method of plant leaf disease detection based on deep learning and convolutional neural network. In *Proc. 6th International Conference on Intelligent Computing and Signal Processing* 816–819 (Xi'an · China, 09 – 11 Apr, 2021).
37. Singh, V. & Misra, A. K. Detection of unhealthy region of plant leaves using image processing and genetic algorithm. In *2015 International Conference on Advances in Computer Engineering and Applications* 1028–1032 (Ghaziabad, India, 19–20 March, 2015).
38. Casado-García, A. et al. LabelStoma: A tool for stomata detection based on the YOLO algorithm. *Comput. Electron. Agric.* **178**, 105751 (2020).
39. Mathew, M. P. & Mahesh, T. Y. Leaf-based disease detection in bell pepper plant using YOLO v5. *Signal Image Video Process.* **16**, 841–847 (2022).

40. Gayathri, S., Wise, D. J. W., Shamini, P. B. & Muthukumaran, N. Image analysis and detection of tea leaf disease using deep learning. In *Proc. International Conference on Electronics and Sustainable Communication Systems*, 398–403 (Coimbatore, India, 2–4 July, 2020).
41. Plant-Village Dataset. https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset: Accessed on Apr. 27, 2023.
42. Sunil, C. K., Jaidhar, C. D. & Patil, N. Systematic study on deep learning-based plant disease detection or classification. *Artif. Intell. Rev.* **56**, 14955–15052. https://doi.org/10.1007/s10462-023-10517-0 (2023).
43. Redmon, J. & Farhadi, A. Yolov3: An incremental improvement," *arXiv preprint* arXiv:1804.02767, (**2018**).
44. Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint* arXiv:2004.10934, (2020).
45. Chen, Z. et al. Plant disease recognition model based on improved YOLOv5. *Agronomy* **12**(2), 365 (2022).
46. Wang, J., Liya, Yu., Yang, J. & Dong, H. DBA_SSD: A novel end-to-end object detection algorithm applied to plant disease detection. *Information* **12**(11), 474 (2021).
47. Peng, Y. & Wang, Yi. Leaf disease image retrieval with object detection and deep metric learning. *Front. Plant Sci.* **13**, (2022).
48. Wang, H. et al. Plant disease detection and classification method based on the optimized lightweight YOLOv5 model. *Agriculture* **12**(7), 931 (2022).
49. Sunil, C. K., Jaidhar, C. D. & Patil, N. Tomato plant disease classification using Multilevel Feature Fusion with adaptive channel spatial and pixel attention mechanism. *Expert Syst. Appl.* **228**, 120381. https://doi.org/10.1016/j.eswa.2023.120381 (2023) (**ISSN 0957-4174**).
50. Sunil, C. K., Jaidhar, C. D. & Patil, N. Binary class and multi-class plant disease detection using ensemble deep learning-based approach. *Int. J. Sustain. Agric. Manag. Inf.* **8**(4), 385–407 (2022).

Acknowledgements

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number 223202.

Author contributions

Conceptualization, Y.A. and M.K.; methodology, M.H.S. and M.I.; software, M.A.; validation, Y.A. and M.A.; formal analysis, M.K.; investigation, M.I.; resources, M.A.; data curation, M.A.; writing—original draft preparation, M.K. and M.H.S; writing—review and editing, M.H.S. and M.A; funding acquisition, Y.A. All authors have read and agreed to the published version of the manuscript.

Funding

Deputyship for Research & Innovation, Ministry of Education, Saudi Arabia, 223202.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Y.A.