automation

Article

Intelligent Tomato Leaf Disease Detection and Automated Spray Prescription Using YOLOv9: A Smart Agriculture Approach

Shahab Ul Islam [1,*], Giampaolo Ferraioli [1], Ghassan Husnain [2,*], Abdul Waheed [2,*] and Vito Pascazio [3]

[1] Department of Engineering, University of Naples Parthenope, 80143 Naples, Italy; giampaolo.ferraioli@uniparthenope.it
[2] Department of Computer Science, CECOS University of IT and Emerging Sciences, Peshawar 25000, Pakistan
[3] Department of Science and Technology, University of Naples Parthenope, 80143 Naples, Italy; vito.pascazio@uniparthenope.it
* Correspondence: shahabul.islam001@studenti.uniparthenope.it (S.U.I.); ghassan.husnain@cecos.edu.pk (G.H.); abdul@netlab.snu.ac.kr (A.W.)

Abstract

Tomato cultivation is a cornerstone of global agriculture, yet it faces significant challenges from a variety of diseases that can drastically reduce yield and quality. Traditional methods of disease detection, which rely on manual inspection, are labor-intensive, time-consuming, and prone to human error. To address these challenges, this study presents an advanced, automated system for tomato disease detection and spray prescription using an enhanced YOLOv9 (You Only Look Once) model. By leveraging advanced deep learning techniques, the proposed system accurately identifies and detects nine tomato leaf diseases in real-time by making efficient, precise, and accurate decisions. This YOLOv9 model is modified for detecting tomato leaf diseases and optimized for getting higher accuracy and efficiency. The system automatically prescribes the spray based on detected disease, which helps in reducing pesticide use, along with the environmental impact. This system helps in maximizing crop health and yield. After testing the system on the test dataset and real-time images, the results demonstrate the system's accuracy and efficiency, achieving a detection accuracy of 97% and spray prescription accuracy of 94%. Integrating a YOLOv9 with a spray prescription system provides a sustainable, cost-effective solution for managing tomato plant diseases. Implementing this system on edge devices paves the way for more extensive precision agriculture applications. By integrating advanced technology with real-world agricultural needs, this work makes a contribution and a global effort to ensure food security and ecological farming practices.

Keywords: tomato leaf; disease detection; spray prescription; YOLOv9; leaf disease detection

Check for updates

Academic Editor: Shuai (Steven) Li

Received: 27 September 2025
Revised: 19 December 2025
Accepted: 26 December 2025
Published: 25 February 2026

1. Introduction

Tomato (*Solanum lycopersicum*) is one of the most economically significant horticultural crops in the world, with an annual global production of over 186 million tonnes and a market value of over $190 billion [1]. Tomatoes are a major contributor to global food security and human nutrition because they are a rich source of vital nutrients like lycopene, β-carotene, vitamins C and E, and many other bioactive compounds [2]. However, persistent challenges from bacteria, fungi, and other viral diseases in tomato plants result in a crop yield loss of 10–25% globally, which is equivalent to the annual economic damage of $19–48% [3].

In tomato cultivation, conventionally, disease management is totally dependent upon the calendar-based fungicide applications, which results in ecological pollution, excessive use of pesticides, a potential development of pathogen resistance, and an increased cost of production [4]. According to the Food and Agriculture Organization, each year, over 2 million tonnes of pesticides are used worldwide, a remarkable portion of which is used for tomato production systems [5]. This approach of excess usage of pesticides through unnecessary chemical applications not only elevates the health and environmental concerns but also imposes a significant economic burden on the farmers [6].

For implementing targeted and timely interventions, it is crucial to detect plant diseases accurately in early stages. An early detection of plant disease can significantly minimize crop losses while reducing the use of chemicals. Traditional methods of plant disease identification involve a visual assessment by experts or laboratory analysis. These methods are costly, time-consuming, and difficult to implement for large fields [7]. In the early stages of plant diseases, visual identification is very complicated due to similarities among symptoms that are caused by abiotic stresses, nutritional deficiencies, or different pathogens [8]. These constraints often result in inaccurate diagnoses, due to which diseases spread through the crops before proper remedies can be deployed.

To revolutionize the accurate management of plant diseases, computer vision and Artificial Intelligence play a very important role. Due to the advancement in these fields, outstanding opportunities have been introduced to detect and manage plant diseases accurately and timely. In particular, Convolutional Neural Networks (CNNs) have shown significant potential to identify visible patterns of plant pathologies across varying crop systems [9]. In real-time agriculture applications, the evolution of deep learning algorithms, especially the YOLO (You Only Look Once) algorithm family, has gained remarkable attention due to its outstanding performance regarding speed and accuracy [10,11]. In early 2024, YOLOv9 was released with its significant and enhanced Generalized Efficient Layer Aggregation Network (GELAN) architectures and innovative Generalized Efficient Layer Aggregation Network (GELAN) architectures [12].

Current research highlights numerous approaches for detecting plant diseases. This work started with a detailed literature survey on evolving techniques for accurate detection of plant diseases, thus understanding the current knowledge on the subject by establishing a foundation.

Recent advancements in deep learning, specifically the evolution of the YOLO framework, have significantly enhanced the accuracy of tomato disease detection in early stages. Most of the researchers worked on modifying the architecture of the YOLO family to improve accuracy in detecting the early-stage symptoms and small lesions. The researchers in [13] modified the architecture of YOLOv8 and YOLOv9 in [14]. In both works, the authors combined the multiscale attention mechanism, the lesion-focused detection heads, and data augmentation of specific diseases to get a higher mean average precision (mAP) by up to 13.7%. In [15], the authors perform a comparative analysis, which further reveals that the YOLOv9 performed better than YOLOv8, EfficientDet, and Faster R-CNN in both speed and accuracy (94.3% mAP). Other researchers have achieved higher accuracies (88.7–93.5%) by using architectures like ResNet-152 [16], EfficientNet [17], and MobileNetV3 [18], or by refining Faster R-CNN for controlled environments [19].

Apart from getting higher disease detection accuracy, most of the findings focus on the implementation of these models on resource-constrained edge devices like drones, robotic arms, or mobile platforms. To deploy these models on edge devices in a real-time environment, the researchers implemented quantization, pruning, and knowledge distillation techniques on YOLOv7. By applying these compression techniques, the authors achieved high frame rates on Raspberry Pi hardware while reducing the parameter counts by 73%. These optimized algorithms on the edge devices, which are enhanced with environmental sensors, ensure reliable performance by adjusting the image preprocessing based on variable lightning and weather conditions [20,21].

Significant research tends to focus on integrating detection models into comprehensive Decision Support Systems (DSS) and autonomous agricultural hardware. By combining YOLO-based detection with multispectral imagery, systems can now identify subtle spectral changes 3–4 days before visible symptoms appear, significantly reducing infection rates [22]. Integrated frameworks have been deployed via robotic sprayers [23] and drone-based multispectral mapping [24] to enable variable-rate chemical applications. Field validation of these systems determined that the use of pesticide chemicals was reduced by 37% when compared to the conventional methods. This automated system can control and enhance pest control, leading to a positive return through better crop quality and cost savings.

Despite these advances, in the domain of automated tomato disease management, numerous critical research gaps still exist. In the most recent studies, there is a significant advancement in tomato disease detection algorithms, but their integration with the automated spray prescription systems for detected diseases remains limited. Only a few studies show successful integration of the detection outcomes with specific recommendations like the type of disease and its severity, along with the environmental conditions. Most of the proposed studies showed a high accuracy under controlled conditions, but their performance reduced during variable field environments where there are complex backgrounds, inconsistent light, and varying disease indications. Real-time agricultural applications, specifically installed on edge devices, need to process the images fast enough for quick and timely decision-making. For such applications, capturing high-resolution images becomes a challenge in dynamic field environments to maintain the balance between computational efficiency and detection accuracy.

In this research work, we implemented YOLOv9 for detecting and classifying nine distinct tomato leaf diseases: "Early Blight", "Late Blight", "Septoria Leaf Spot", "Leaf Mold", "Target Spot", "Spider Mites", "Yellow Leaf Curl Virus", and "Mosaic Virus", as well as "Healthy" leaves. The system is integrated with an automated spray prescription module that recommends appropriate interventions based on detected pathologies, facilitating targeted application of crop protection products.

The main goals of this research work are as follows:

1. To develop and validate a YOLOv9-based detection system for nine tomato leaf conditions with high accuracy (>95% mean Average Precision) and real-time processing capabilities (<50 ms per frame on standard hardware).
2. To develop an integrated system that automatically prescribes the treatment based on the disease detected.
3. To validate the performance of the system across various conditions, like leaf angles, complex backgrounds, and variable lighting.
4. To evaluate the possible reduction in usage of pesticides attained using this integrated approach as compared to the traditional calendar-based spray systems.
5. To assess the economic viability of the system through cost-benefit analysis, considering implementation costs versus savings in crop protection products and potential yield preservation.

The novel contribution of this work includes the following:

1. Development of a tomato disease detection system trained with a dataset of more than 10,000 images with variable lighting conditions, different growth stages, and severities of nine tomato leaf diseases.
2. Integration of the tomato disease detection module with the automated spray prescription module, which will automatically prescribe the spray based on the disease detected.

3. Implementation of advanced data augmentation techniques and transfer learning strategies to enhance model robustness across variable field conditions.
4. Development of a lightweight model variant optimized for deployment on edge computing devices.
5. Extensive field validation across multiple growing regions to assess system performance under real-world agricultural conditions.

The organization of this research work is as follows:

Section 1 describes the introduction and background literature on the use of deep learning applications in plant disease detection, with a specific focus on the YOLO family and tomato pathologies. Section 2 provides a detailed description of the research flow, along with the dataset collection and preparation, model architecture, performance evaluation, and the spray prescription module. Section 3 provides the experimental results and findings of our integrated system. Section 4 provides a detailed discussion of our findings, comparison with existing approaches, the limitations of the system, and future research directives. Finally, Section 5 concludes the paper.

2. Materials and Methods

This research work starts with a linear workflow for detecting and classifying the tomato leaf diseases using the YOLOv9 model and spray prescription based on the detected disease. The workflow consists of the five stages, including data acquisition and preparation, the model architecture, disease detection and performance evaluation, and spray prescription, establishing a standardized approach for model development and performance evaluation. This approach provides a robust and accurate solution that is suitable for real-world precision agriculture applications. All the steps in the methodology are interconnected, and each contributes to the overall performance of this work, as shown in Figure 1.

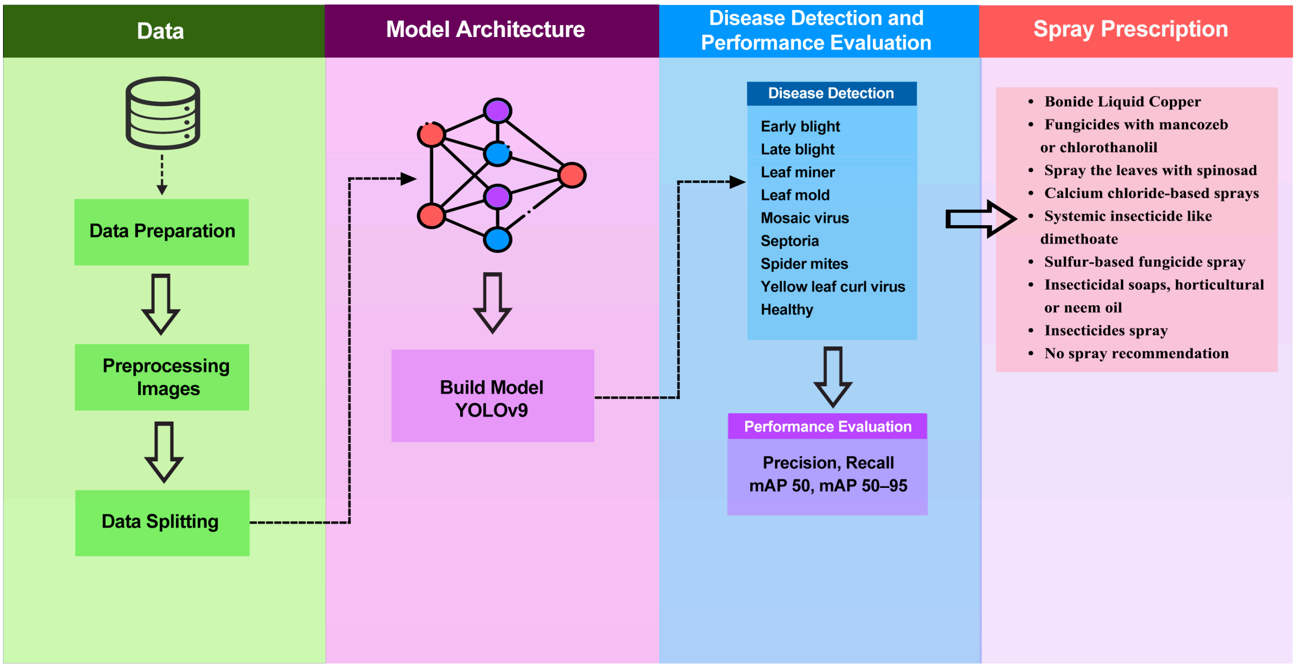


Figure 1. Research flow.

2.1. Data Acquisition and Preprocessing

Images of tomato leaves were collected from publicly available datasets (RoboFlow Universe) and field-captured photographs using a drone camera, focused on common tomato diseases, including healthy leaves. The size of our dataset comprises 10,047 images with balanced distribution across classes, as shown in Table 1. To match YOLOv9's input requirements, the images were resized to 640 × 640 pixels. For improving the generalization, augmentation techniques like rotation, flipping, scaling, and HSV adjustments are applied. The dataset is divided into training (70%), validation (15%), and testing (15%) sets. The unlabeled images, which we have obtained through a drone camera, have been labeled using the LabelImg tool. For loading the images and labels from the repository, the provided model automatically performs the data preprocessing for the classification tasks. For converting the images from BGR to RGB format, OpenCV (cv2) is used, which reads the images from the specified paths and converts them, showing consistency across the images.

Table 1. Class label plant.

Class Label	No. of Images	Image Size
Early Blight	3000	640 × 640 pixel
Late Blight	1200	640 × 640 pixel
Leaf Miner	1234	640 × 640 pixel
Leaf Mold	4532	640 × 640 pixel
Mosaic Virus	1234	640 × 640 pixel
Septoria Leaf Spot	432	640 × 640 pixel
Spider Mites	43	640 × 640 pixel
Yellow Leaf Curl Virus	124	640 × 640 pixel
Healthy	432	640 × 640 pixel

In the preprocessing of the data, we achieved uniform image formatting, which is RGB format, proper organization of images along with label paths, and preparation of data for training YOLOv9. For image preprocessing, YOLOv9 draws bounding boxes around the regions of interest (ROIs) within an image. For tomato disease detection in the tomato leaves, this is considered a very important step. The process involved going through the initial five training images and their labels. For each step, the model displayed the image and drew a bounding box on it. To ensure these boxes were accurately placed and sized on the actual image, the model converted their relative coordinates into absolute pixel values using the image's width and height.

For improving the generalization of the images and enhancing the robustness, augmentation techniques like rotation, flipping, scaling, and brightness adjustments are applied to the dataset. This strengthens the model's generalizing power in simulating real-world varied scenarios.

2.2. Model Architecture

A neural network is basically a computational model that is good at finding patterns and making decisions from data. It is made up of interconnected "neurons" in different layers that process the input data and generate output based on the input provided. There is an input layer to get the data, a bunch of "hidden" layers that do most of the complex thinking and transformations, and an output layer for the final result. These hidden layers are key; the more you have, the better the network can learn intricate patterns.

$$z = W\alpha + b \tag{1}$$

The equation for the input layer is given in Equation (1), where

- z is the input passed to the activation function;
- W is the weight matrix for the current layer;
- α is the output from the previous layer.

α is the activation function (e.g., Sigmoid, ReLU, or Tanh) that processes z.

$$\alpha = \sigma(z) \tag{2}$$

$$\hat{y} = W^L \alpha^{L-1} + b^L \tag{3}$$

Equation (2) shows that $\hat{y}$, which is the final output or the network prediction, is calculated by operating the activation function on the weighted sum of the output of the last hidden layer. Equation (3) defines the output using the weight matrix (W^L) of the final layer and the activation output (α^{L-1}) of the last hidden layer with the calculated bias incorporated. YOLOv9 is designed to manage complex data while providing real-time object detection. Its configuration, including its architecture and custom data loader, facilitates efficient image processing. By taking advantage of the underlying principles and features extracted from an autoencoder model, YOLOv9 ensures optimal object detection performance.

The YOLOv9 utilized consists of a default backbone (CSPDarknet), neck (PANet), and detection head, as shown in Figure 2. The anchor boxes are modified to suit smaller leaf lesion sizes.

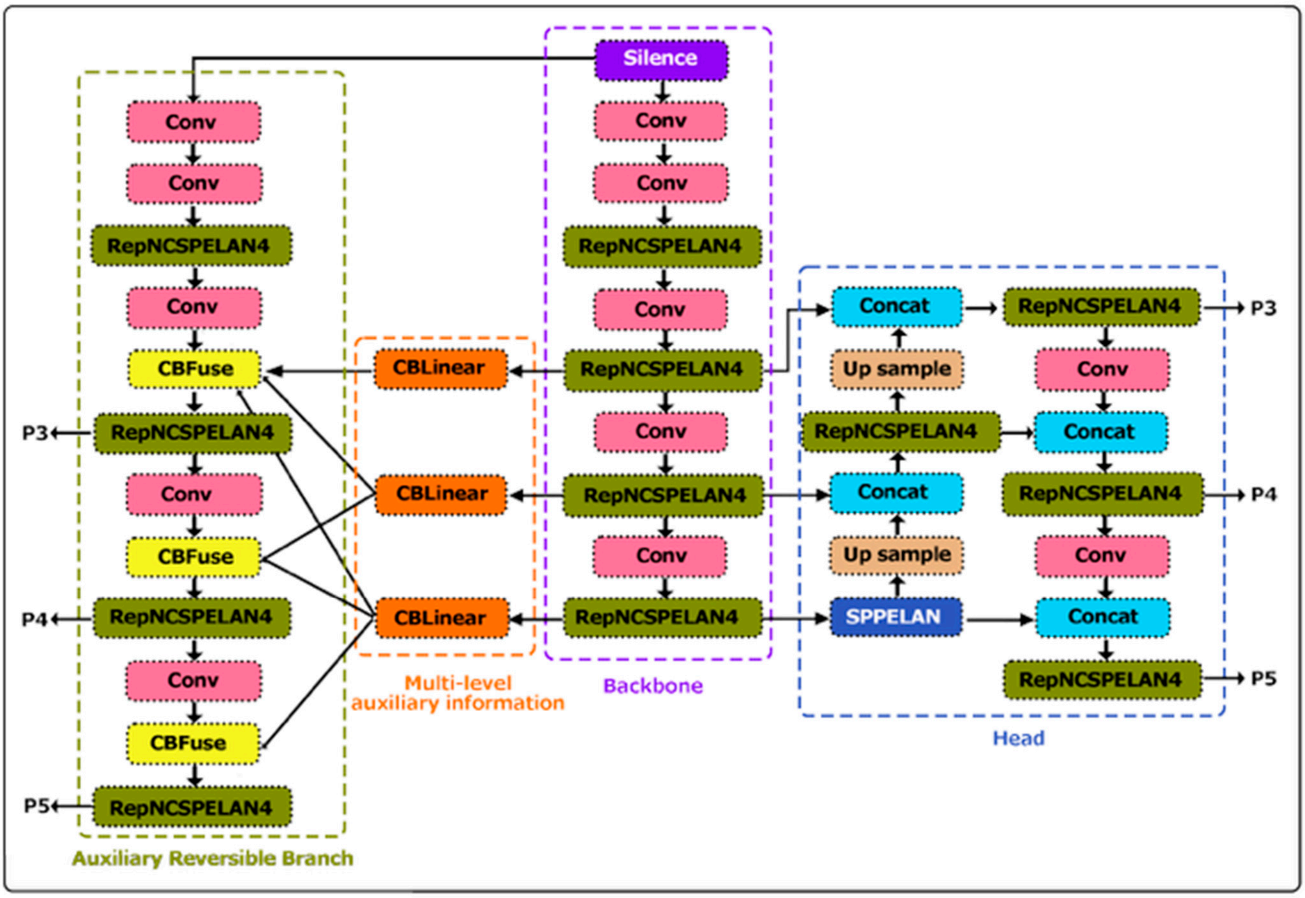


Figure 2. Proposed YOLOv9-based architecture.

For preparing the images for robust and efficient preprocessing, the data loader is used for performing essential preprocessing steps, which resize the images to uniform dimensions (128 $\times$ 128) using transforms. The data loader normalizes and resizes the pixel values to prevent feature control.

The attention mechanisms, like (CBAM modules), are enabled for improved feature extraction. The configuration of the model is as follows:

- The pre-trained COCO weights are used for model initialization.
- Batch size: 16.
- Optimizer: Adam with momentum (0.937) and weight decay (0.0005).
- Learning rate of 0.01 along with cosine annealing scheduler.
- No. of Epochs: 150.
- Hardware: Trained on NVIDIA GPUs (e.g., RTX 3090/4090) with CUDA acceleration.

2.3. Disease Detection and Model's Performance Evaluation

To assess the performance of the YOLO models, a performance evaluation is very important to know about the efficiency and accuracy of the model. To assess the performance of the YOLO models, the following key metrics are used: precision score, recall score, mAP0.5, and mAP0.5–0.95. The precision score measures the proportion of the positive predictions by the model that were actually correct (the correct diseases it detected). In contrast, the recall score calculates the proportion of all actual diseases that the model detected successfully. Mean Average Precision at IoU (mAP) is used to assess this performance. The Intersection over Union (IoU) metric itself quantifies the spatial accuracy by computing how well a predicted bounding box overlaps with the ground truth bounding box.

mAP (0.5–0.95) provides a detailed evaluation of the detection accuracy across a range of overlap thresholds, with a higher score indicating superior performance. For the performance evaluation of our model, we used precision, recall, F1-score, mAP (at both IoU thresholds), and a Confusion Matrix for error visualization. Fundamentally, mAP is the average of the Average Precision (AP) values computed across various IoU thresholds, where AP is determined by combining the Precision-Recall curve. To generate the Precision-Recall curve, the precision and recall at various confidence levels are measured systematically. To get an overall sense of performance, the area under this curve using a numerical method, like the trapezoidal rule, is then approximated. The Mean Average Precision is computed for a more robust evaluation. This includes computing average precision across different IoU thresholds (like 0.5 or a range from 0.5 to 0.95) and then taking the average of those AP values to calculate the final mAP score.

For disease detection of tomato leaves, the performance evaluation metrics are listed below (4)–(7):

$$\text{precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{F1-score} = \frac{(2 \times TP)}{2 \times TP + FN + FP} \tag{6}$$

$$\text{IoU} = \frac{(TP)}{TP + FN + FP} \tag{7}$$

where

- *FP*: False Positives (incorrectly predicted positive cases).
- *TP*: True Positives (correctly predicted positive cases).
- *FN*: False Negatives (incorrectly predicted negative cases).

2.4. Automated Spray Recommendation Module

The spray prescription module serves as the intelligent brain of a precision agriculture system, translating the raw output from disease detection models, such as YOLOv9, into actionable, site-specific spraying instructions. After the YOLOv9 model accurately identifies and localizes tomato diseases by providing bounding box coordinates, class labels, and confidence scores, the prescription module takes over to determine what, how much, and exactly where to spray. In this research work, we have collected data regarding sprays used for tomato plant treatment, as shown in Table 2.

Table 2. Spray list.

No.	Spray
1	Bonide liquid copper
2	Use fungicides with mancozeb or chlorothalonil
3	Spray the leaves with spinosad
4	Calcium chloride-based sprays
5	Spraying of systemic insecticides like dimethoate
6	Spray the leaves with sulfur-based fungicide
7	Spray the leaves with insecticidal soaps, horticultural oils, or neem oil
8	Spray the leaves with insecticides
9	No spray recommendation

This module simply prescribes the spray based on the detected tomato plant disease. We created a lookup table linking diseases to recommended sprays (e.g., copper-based fungicides for bacterial spots and chlorothalonil for blight).

3. Results

This section presents experimental findings derived from applying the YOLOv9 model for tomato plant disease detection and subsequent disease assessment, leading to spray prescription. The performance of the trained YOLOv9 model is first evaluated, followed by a detailed analysis and the derived practical recommendations. YOLOv9 achieved the following evaluation metrics on the test set. YOLOv9 underwent a detailed performance evaluation, and the above-discussed metrics summarize its performance on the test dataset. The normalized confusion matrix generated summarizes the performance of YOLOv9 on tomato leaf disease detection in Figure 3.

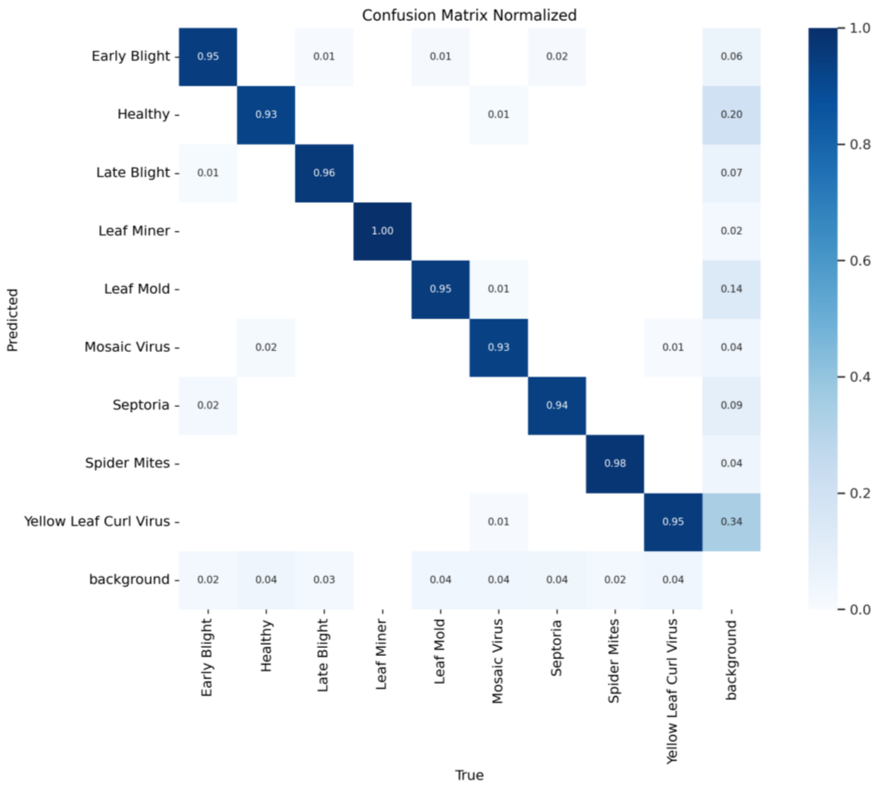


Figure 3. Confusion matrix normalized.

This confusion matrix included nine classes with an additional "background" class. The background class suggests that the model is detecting background (soil and background plants). The results in the confusion matrix show that our model achieves a higher classification accuracy, as evidenced by the higher values focused on the main diagonal, demonstrating a True Positive Rate (Recall) for each class. Particularly, the model showed an excellent performance in detecting "Leaf Miner", "Late Blight", "Early Blight", and "Leaf Mold" with a higher recall rate. The major areas of misclassification are generally low. However, the largest misclassification occurs for "Yellow Leaf Curl Virus", which is often confused with the "background". Figures show that 34% of the time, the model predicted the background as Yellow Leaf Curl Virus. Similarly, the "background" class itself shows notable confusion, with 20% of true "Healthy" samples being incorrectly classified as "background" and 34% of true "Yellow Leaf Curl Virus" being classified as "background". Overall, the matrix confirms that YOLOv9 is highly effective at distinguishing between the various specific disease types.

The YOLOv9 model, specifically trained for the identification of diseases on tomato leaves, demonstrated robust performance across the test dataset. Quantitative evaluation was conducted using standard object detection metrics, including precision (P), recall (R), and F1-score and mean Average Precision (mAP@50). The performance of YOLOv9 is assessed with standard segmentation metrics, including qualitative visual assessments. We also report the robustness and efficiency of how well YOLOv9 and the spray prescription module perform together to detect disease in tomato leaves and prescribe the spray based on disease detection.

Beyond basic testing, we thoroughly evaluated the model by using a range of IoU thresholds, from a moderate 0.5 to a very demanding 0.95. This approach gave us a much clearer picture of its performance, especially under conditions requiring very precise bounding box placement. The training visualization graph in Figure 4 clearly shows the mAP scores at these varied IoU levels, proving YOLOv9's impressive ability to maintain strong performance even when precise object localization is critical. Notably, YOLOv9 surpassed its predecessors, achieving an average mAP of 97% at a 0.5 IoU threshold and an excellent 93% when averaged across the 0.5–0.95 IoU range.

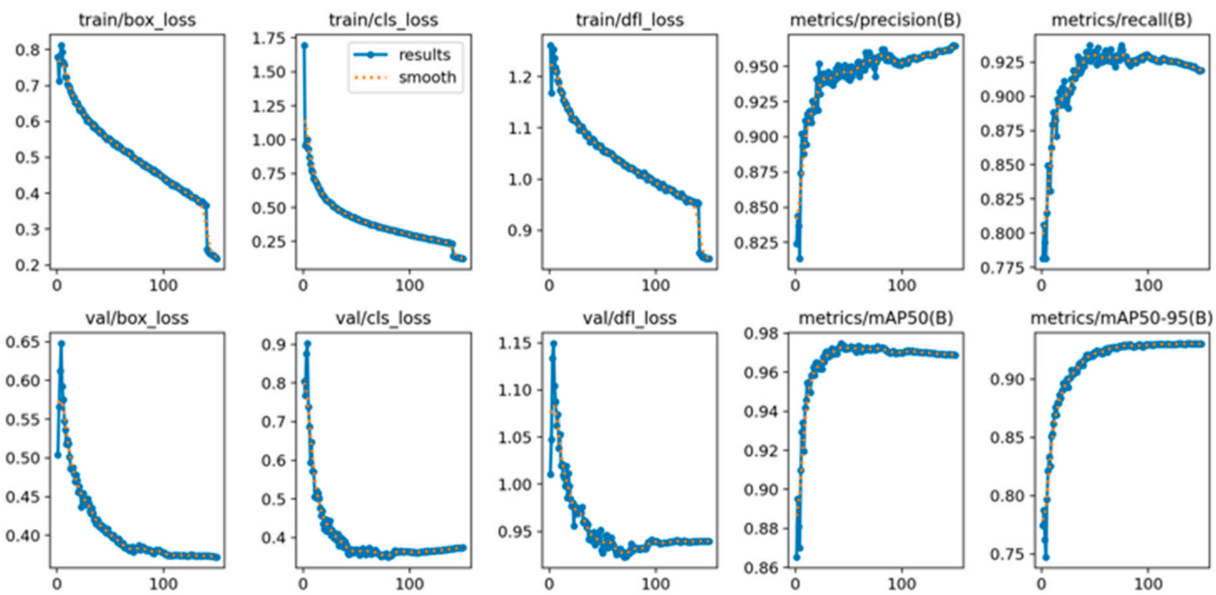


Figure 4. Training visualization results.

It was good at recognizing other diseases with high accuracy, but it recognized the leaf curl virus disease with an accuracy of 83.2%. This network showed excellent performance, as compared with both YOLOv7 and YOLOv8 by approximately 3–4% in terms of mAP. Crucially, it reduced misclassification errors, a benefit particularly noticeable with diseases that have overlapping symptoms.

Compared to earlier versions, YOLOv9 shows an improved mAP@0.5 value, specifically in handling ambiguous visual symptoms and overlapping disease patterns. Such performance and improvements are very important to real-world agriculture applications that require high reliability and accuracy. These metrics represent a 3–4% improvement over YOLOv7 and YOLOv8, proving the YOLOv9's capabilities in both efficiency and accuracy. The higher mAP scores show its capability in handling challenging disease patterns and overlapping bounding boxes, which is critical for real-world deployment.

In detecting tomato leaf diseases, YOLOv9m proves an outstanding performance by achieving the overall accuracy of 95.8% precision, 92.4 recall, 97% (mAP50), and 93% (mAP50–95) on a validation dataset consisting of 843 images. The inference speed recorded per image was 13.5 ms, among which 0.3 ms was the preprocess speed, while 0.9 ms was the postprocess speed. The efficiency of the model changed across the classes. The "Leaf Miner" class demonstrates the highest accuracy with 99.5% at mAP50, showing a fast and robust disease detection capability. Table 3 shows the model's performance in more detail by mentioning each metric with each class.

Table 3. Model performance.

Class	Images	Precision Accuracy	Recall Accuracy	mAP50	mAP50–95
All	**843**	**0.958**	**0.924**	**0.97**	**0.93**
Early Blight	214	0.956	0.943	0.981	0.955
Healthy	76	0.926	0.874	0.927	0.87
Late Blight	238	0.985	0.944	0.982	0.948
Leaf Miner	199	0.981	0.996	0.995	0.979
Leaf Mold	211	0.947	0.901	0.969	0.927
Mosaic Virus	250	0.974	0.936	0.973	0.958
Septoria	203	0.982	0.88	0.971	0.935
Spider Mites	123	0.976	0.976	0.984	0.967
Yellow Leaf Curl Virus	152	0.896	0.87	0.946	0.832

Table 3 shows the YOLOv9 overall accuracy in detecting the tomato leaf diseases within the dataset provided. The results show the varying performance across different classes. The results also show that the performance of the model is outstanding on the "Leaf Miner" class with a mAP of 97.9%, but with the "Yellow Leaf Curl Virus" class, the model struggled by achieving a mAP of 83.2%. These results reflect the real-time capability, which is very important in real-time application fields that require quick decision-making, for example, disease detection or treatment prescription. The training visualization results are shown in Figure 4, in which we must keep an eye on two main things.

The "train/box_loss" essentially tells us how "on target" our predicted boxes are. When you see this number dropping, it is a clear sign that the model is getting much better at accurately finding and outlining objects. Similarly, the "train/cls_loss" tracks how good the model is at naming what it sees. If this number is also going down, it means the model is improving its identification skills, which is crucial when its job is to spot tomato plant diseases on leaves. The plot on train/dfl_loss adds both classification losses and box into a single metric. A decrease in the train/dfl_loss plot shows the overall improvement in the performance to detect objects. Due to the visual similarities between the lesion patterns of "Septoria Leaf Spot" and "Early Blight", misclassifications may primarily occur. The lesions with comparable colorations, sizes, and shapes make the differentiation process complicated. These similarities make it difficult for the model to differentiate the unique features of diseased leaves from one another, especially in the case of a partially damaged leaf or overlapping environmental conditions. The validation results and the model's results on the test dataset are shown in Figure 5 and Figure 6 respectively.

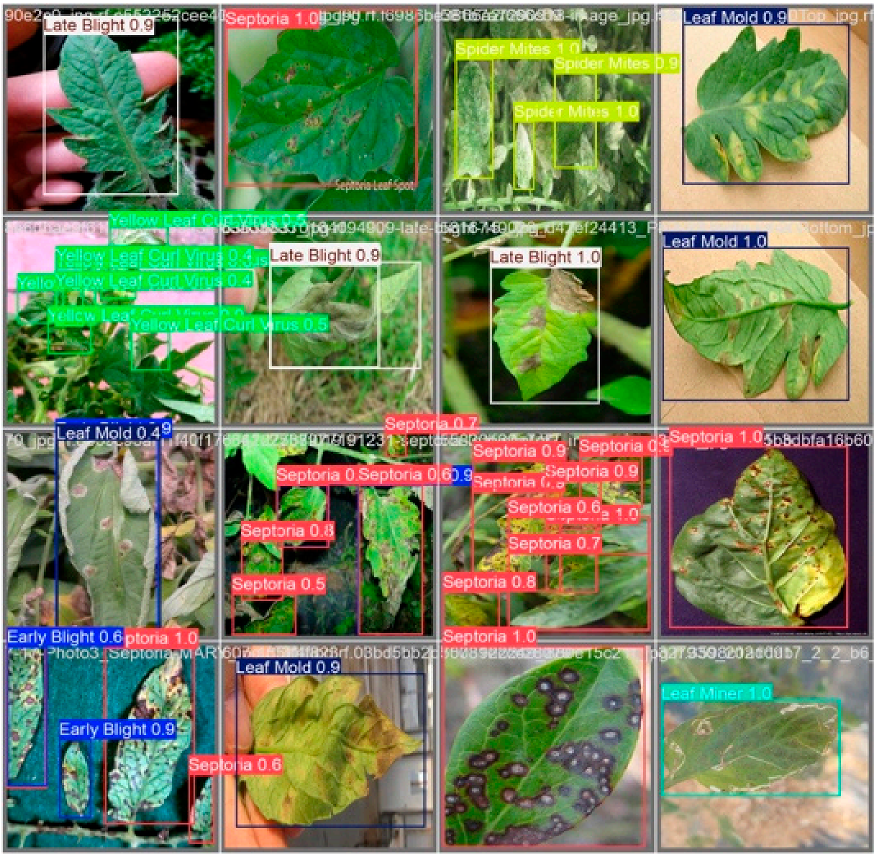


Figure 5. Validation results.

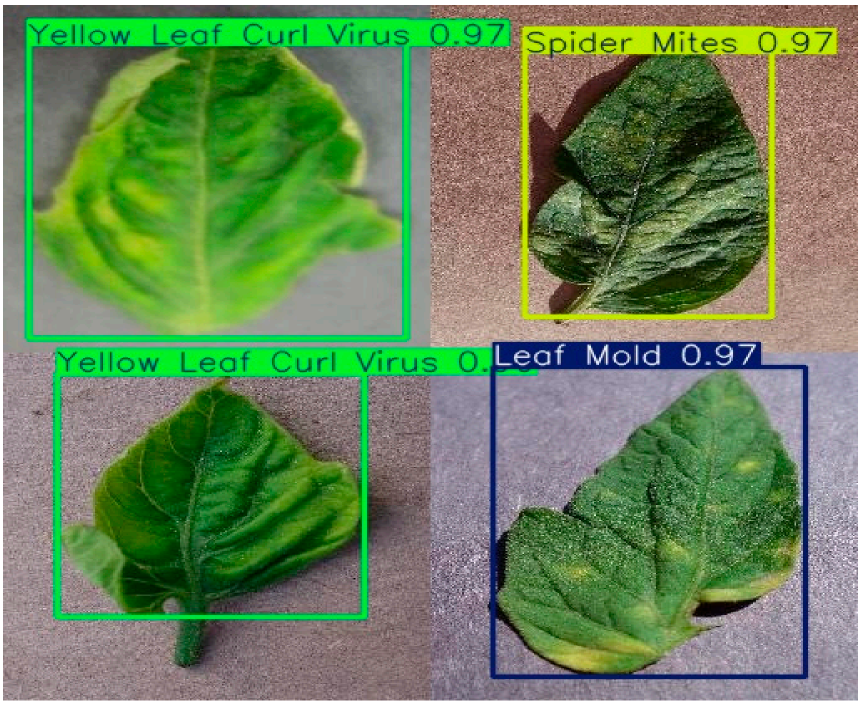


Figure 6. Results on the test data set.

The performance of the model was outstanding in identifying the infected plants with high precision and recall accuracy. For assessing the quality of tomato leaf disease detection, the precision and recall scores for every image were determined at the pixel level. The proportion of true positives is measured by the precision curve, while the recall curve measures the proportion of actual positives that were correctly predicted. The precision-recall curve was 0.98% @mAP50, as shown in Figure 7, while the F1 confidence curve, which is 0.94%, is shown in Figure 8, reflecting the model's ability to correctly identify the tomato leaf diseases and reduce the false positives.

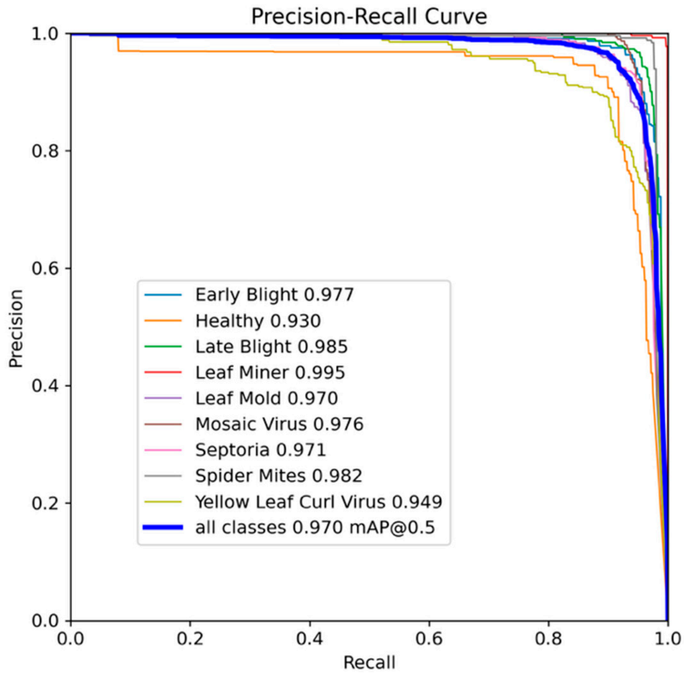


Figure 7. Precision-recall accuracy @mAP50.

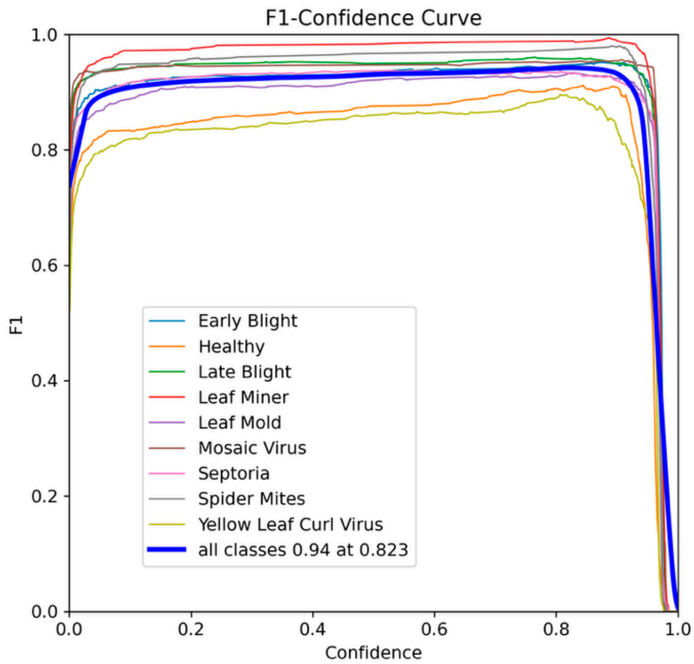
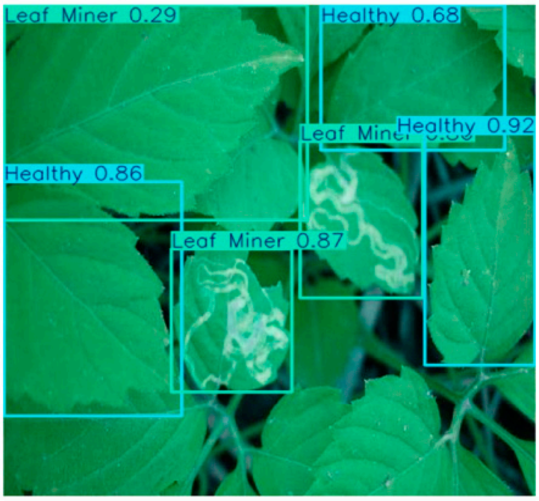


Figure 8. F1 confidence curve.

After the disease is detected correctly by YOLOv9, a spray prescription module that is integrated with the model automatically prescribes the spray based on the detected disease.

Figure A1 in Appendix A shows how the mapping between the diseases and the spray is implemented. The spray prescription result after disease detection is shown in Figure 9 and Figure 10 respectively. The model performs well in identifying the leaf miner disease, but it misclassifies the yellow leaf curl virus.




```
0: 640x640 3 Healthys, 3 Leaf Miners, 17.7ms
Speed: 3.6ms preprocess, 17.7ms inference, 1.2ms postprocess per image at shape (1, 3, 640, 640)
Results saved to runs/detect/predict7
<Figure size 1000x1000 with 1 Axes>
Detected Diseases and Spray Recommendations:
- Healthy: No spray recommendation
- Leaf Miner: Spray the leaves with spinosad it can be an effective option
```

Figure 9. Spray prescription for leaf miner.


```
0: 640x640 4 Late Blights, 17.7ms
Speed: 1.9ms preprocess, 17.7ms inference, 1.3ms postprocess per image at shape (1, 3, 640, 640)
Results saved to runs/detect/predict9
<Figure size 1000x1000 with 1 Axes>
Detected Diseases and Spray Recommendations:
- Late Blight: Spray the infected leaves with Fungicides that contain maneb, mancozeb, chlorothanolil, or fixed copper
```

Figure 10. Spray prescription for late blight.

After the overall performance, it shows that YOLOv9 reduces the workload regarding tomato plant disease detection because it automates the process of disease detection, hence requiring less expertise from the expert. However, in the case of the yellow leaf curl virus, where the model's accuracy is reduced by misclassification, the model needs improvement by providing extra augmentation techniques or domain-specific features such as spectral imaging.

4. Discussion

The application of YOLOv9 for tomato disease detection, coupled with an automated spray prescription module, represents a significant advancement in precision agriculture. This integrated system offers a powerful tool for timely and targeted intervention, contributing to enhanced crop health, reduced chemical usage, and improved yield. Our YOLOv9 model demonstrated promising capabilities in accurately identifying various tomato diseases. The architecture's inherent strengths, such as high object detection accuracy and its abilities in real-time processing, are particularly well-suited for agricultural applications where rapid diagnosis is crucial. By taking the benefits of the advanced detection and feature extraction mechanisms of YOLOv9, the system can differentiate between healthy and diseased plant parts, and even distinguish between different types of diseases, often with high precision and recall. This is a considerable improvement over traditional inspection methods that are conducted manually, which require more time and labor cost, and are prone to human error and subjectivity. The ability to localize lesions and identify disease types enables more precise diagnosis, which is a critical first step towards effective management.

The seamless integration of the disease detection module with an automated spray prescription system is a key strength of this work. Once a disease is identified by YOLOv9, the system can automatically trigger a recommendation for the appropriate pesticide or treatment, along with the recommended dosage and application method. This automation streamlines the decision-making process for farmers, minimizing delays and ensuring that interventions are applied at the optimal time.

4.1. Comparative Analysis

A comparative analysis of the past algorithms that have been employed for tomato leaf disease detection is shown in Table 4. It summarizes each algorithm used by the authors along with the performance metrics like precision, recall, and mAP, and the integration of models into spray prescription for real-time applications.

Table 4. Comparison of our model with different detection models.

Approach	Precision	Recall	mAP50	Spray Prescription
DM-YOLOv9	92.5%	86.8%	86.4%	No
DMW-YOLOv8	95.4%	83.6%	88.6%	No
YOLOv8n	76.1%	77.6%	82.4%	No
BED-YOLO (YOLOv10)	87.2%	89.1	91.3	No
Our Work	**95.8**%	**92.4**%	**97**%	**Yes**

For automated tomato leaf disease detection, these studies adopt various approaches for developing deep learning models to enhance the accuracy of the models through architectural improvements. Notably, by integrating the YOLOv9 model with a spray prescription module, our study distinguishes itself, enabling real-time, automated pesticide application. The above comparison indicates the performance of YOLOv9 in different conditions, like variant light conditions and the presence of small leaves. This comparison also highlights the practical implications of YOLOv9 in precision agriculture. Our proposed work has shown outstanding performance in terms of detecting tomato leaf diseases and automatically prescribing a spray for tomato leaves based on the detected disease. It is important to note that other models, such as DM-YOLOv8 [25], DMW-YOLOv8 [26], YOLOv8n [27], and BED-YOLO [28], have their merits, but this research work indicates a comparable level of performance, as shown in Table 4. The visualization of comparative analysis with the previous models with respect to precision, recall, and mAP metrics is shown in Figure 11.

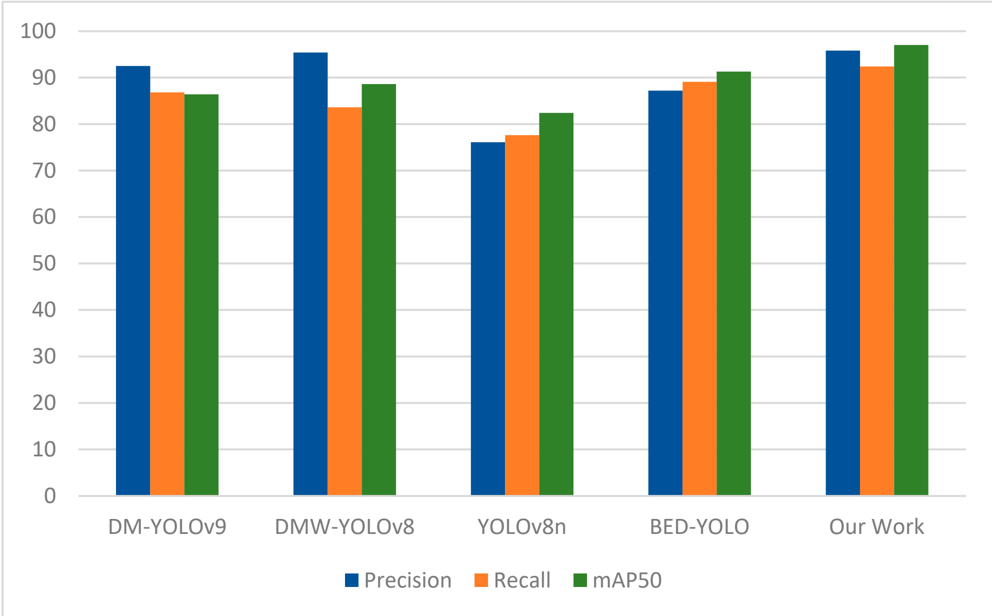


Figure 11. Comparative analysis with previous models based on precision, recall, and accuracy.

4.2. Limitations of Our Work

Despite the significant advantages, several limitations need to be acknowledged to ensure a comprehensive understanding of the system's current capabilities and future potential. The performance of the YOLOv9 model is heavily reliant on the quality, quantity, and diversity of the training dataset. The model's accuracy might decrease when encountering less common or atypical disease symptoms not adequately represented in the training data. Early-stage disease detection is very challenging in every plant. Variable environmental conditions, like variable lightning conditions (direct sunlight, shadows, etc.), background clutter (other plants, soil, etc.), and leaf occlusion, can considerably affect the detection accuracy in real-world field applications. The complexity of these diverse environments may not fully be captured in the training dataset. These visual clues (symptoms' appearance and disease occurrence) can geographically vary due to pathogen strains and multiple environmental factors. A model trained on data from one region might not perform well in another region. Implementing YOLOv9 on edge devices in real-time requires significant computational power. On edge devices, the resource constraints might lead to a trade-off between inference speed and detection accuracy. The spray prescription module in this research work might offer a static recommendation of spray solely based on the disease detected. It may not dynamically adapt to other crucial factors, such as disease severity, the specific growth stage of the tomato plant, and local weather conditions. The cost of implementing such a system, including hardware (cameras, computing units for edge deployment, and automated sprayers) and software development, might be very expensive for small-scale farmers. It will be difficult for most of the farmers to operate and maintain this system, as this system might require a certain level of technical expertise.

4.3. Future Direction

To further enhance the robustness and utility of this system and to address the above-discussed limitations of this system, future research work should focus on training the model on a wider range of datasets that consist of a wider range of environmental conditions, disease severities, and co-occurring diseases. Future research needs to be focused on YOLOv9 optimization techniques while implementing resource-constrained edge devices for real-time processing. For more dynamic and precise treatment recommendations, future work needs to integrate real-time environmental data (temperature, humidity, and rainfall) and the information on the plant growth stage. Additionally, before the spray prescription module, the severity of the disease should be determined, and then the spray prescription module should automatically prescribe the spray and quantity of spray according to the disease severity. By addressing these points, the YOLOv9-based tomato disease detection and spray prescription system can evolve into an even more powerful and indispensable tool for sustainable and efficient tomato cultivation.

5. Conclusions

This research work provides a highly effective and robust system to detect tomato plant diseases in early stages and automatically prescribe the spray prescription based on the disease detected. This research work addresses a critical need in modern agriculture. The integration of the YOLOv9-based disease detection model with the automated spray prescription module represents a significant step forward in precision agriculture for tomato cultivation. The system achieves a precision score of 95.8%, a recall score of 92.4%, and an impressive mAP50 of 97%, clearly demonstrating the system's accuracy, efficiency, and reliability in real-world agricultural applications. The ability of this system to accurately detect diseases, along with its capacity to automatically prescribe the spray based on detected disease, offers profound implications for sustainable and efficient farming practices. By reducing false positives, the system aims to minimize the use of pesticides, which is helpful in reducing the operational cost for farmers and the environmental impact. This approach provides a powerful tool for farmers to make timely and informed decisions in managing tomato plant cultivation. This research work not only validates the effectiveness of advanced deep learning models like YOLOv9 in complex agricultural applications but also provides a tangible solution for enhancing food security and promoting sustainable agricultural practices. Future research could explore dynamic spray prescription modules that adapt to environmental conditions, disease severity, and economic factors, as well as the potential for integration with robotic platforms for fully autonomous disease management and targeted spraying.

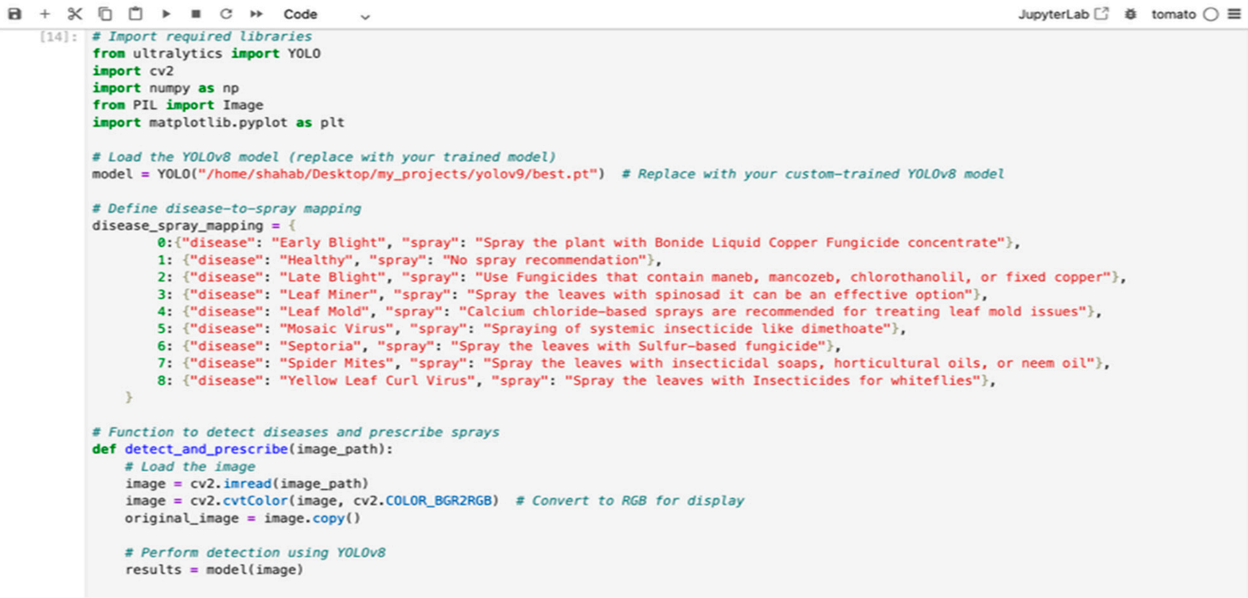
Author Contributions: S.U.I. conceived the presented idea. S.U.I. designed and developed the model and the computational framework and analyzed the data. G.F. and V.P. encouraged and supervised the findings of this work. G.H. and A.W. reviewed this work. All authors discussed the results and contributed to the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset we have used in this research work has obtained from the RoboFlow Universe. The **personal-enkcf** account on Roboflow Universe is primarily associated with the "Tomato Leaf Disease Detection" dataset and project. This public dataset is designed for training computer vision models to identify various tomato leaf diseases. The link to this dataset is as follows: https://universe.roboflow.com/personal-enkcf/tomato-leaf-disease-detection-gyozv (accessed on 20 December 2025).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

Appendix A. Supplementary Technical Materials




```python
# Import required libraries
from ultralytics import YOLO
import cv2
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

# Load the YOLOv8 model (replace with your trained model)
model = YOLO("/home/shahab/Desktop/my_projects/yolov9/best.pt")  # Replace with your custom-trained YOLOv8 model

# Define disease-to-spray mapping
disease_spray_mapping = {
        0:{"disease": "Early Blight", "spray": "Spray the plant with Bonide Liquid Copper Fungicide concentrate"},
        1: {"disease": "Healthy", "spray": "No spray recommendation"},
        2: {"disease": "Late Blight", "spray": "Use Fungicides that contain maneb, mancozeb, chlorothanolil, or fixed copper"},
        3: {"disease": "Leaf Miner", "spray": "Spray the leaves with spinosad it can be an effective option"},
        4: {"disease": "Leaf Mold", "spray": "Calcium chloride-based sprays are recommended for treating leaf mold issues"},
        5: {"disease": "Mosaic Virus", "spray": "Spraying of systemic insecticide like dimethoate"},
        6: {"disease": "Septoria", "spray": "Spray the leaves with Sulfur-based fungicide"},
        7: {"disease": "Spider Mites", "spray": "Spray the leaves with insecticidal soaps, horticultural oils, or neem oil"},
        8: {"disease": "Yellow Leaf Curl Virus", "spray": "Spray the leaves with Insecticides for whiteflies"},
    }

# Function to detect diseases and prescribe sprays
def detect_and_prescribe(image_path):
    # Load the image
    image = cv2.imread(image_path)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  # Convert to RGB for display
    original_image = image.copy()

    # Perform detection using YOLOv8
    results = model(image)
```

Figure A1. Spray mapping logic.

The programming script in Figure A1, which is implemented in Python version 3.9 utilizing the YOLOv9 architecture, describes the technical overview of spray prescription logic based on the detected disease. The core of the prescription mechanism is a structured mapping dictionary, disease_spray_mapping, which links specific diagnostic indices (0–8) to targeted treatment protocols. This script serves as a transition from computer vision output to actionable agricultural advice.

References

1. FAO. *FAOSTAT Statistical Database*; Food and Agriculture Organization of the United Nations: Rome, Italy, 2024. [CrossRef]
2. Savary, S.; Willocquet, L.; Pethybridge, S.J.; Esker, P.; McRoberts, N.; Nelson, A. The global burden of pathogens and pests on major food crops. *Nat. Ecol. Evol.* **2019**, *3*, 430–439. [CrossRef]
3. Schieffer, J.; Dillon, C. The economic and environmental impacts of precision agriculture and interactions with agro-environmental policy. *Precision Agric.* **2015**, *16*, 46–61. [CrossRef]
4. Thind, T.S. New insights into fungicide resistance: A growing challenge in crop protection. *Indian Phytopathol.* **2022**, *75*, 927–939. [CrossRef]
5. FAO. *Pesticide Use in Global Agriculture: Current Status and Future Prospects*; Food and Agriculture Organization of the United Nations: Rome, Italy, 2023. [CrossRef]
6. Karpinski, I.; Nordheim, S.; Golla, B. Economic and environmental web services for the advancement of precision spraying. In *Precision Agriculture '25*; Wageningen Academic: Wageningen, The Netherlands, 2025; pp. 247–253. [CrossRef]
7. Singh, J.; Kashyap, R.; Bansal, K.; Das, R.; Sandhu, K.; Singh, G.; Saini, D.K. Transition from conventional to AI-based methods for detection of foliar disease symptoms in vegetable crops: A comprehensive review. *J. Plant Pathol.* **2025**, *107*, 1791–1814. [CrossRef]
8. Barbedo, J.G. Plant disease identification from individual lesions and spots using deep learning. *Biosyst. Eng.* **2023**, *216*, 95–107. [CrossRef]
9. Boulent, J.; Foucher, S.; Théau, J.; St-Charles, P.L. Convolutional neural networks for the automatic identification of plant diseases: A systematic review and meta-analysis. *Front. Plant Sci.* **2019**, *10*, 941. [CrossRef] [PubMed]
10. Bhargava, A.; Aasheesh, S.; Prakash, G.O.; Mohammed, H.A.; Peerapong, U.; Monthippa, U. Plant leaf disease detection, classification, and diagnosis using computer vision and artificial intelligence: A review. *IEEE Access* **2024**, *12*, 37443–37469. [CrossRef]
11. Wang, S.; Xu, D.; Liang, H.; Bai, Y.; Li, X.; Zhou, J.; Su, C.; Wei, W. Advances in Deep Learning Applications for Plant Disease and Pest Detection: A Review. *Remote Sens.* **2025**, *17*, 698. [CrossRef]

12. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning what you want to learn using programmable gradient information. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 15–20 June 2025; pp. 8934–8945. [CrossRef]
13. Shen, Y.; Yang, Z.; Khan, Z.; Liu, H.; Chen, W.; Duan, S. Optimization of Improved YOLOv8 for Precision Tomato Leaf Disease Detection in Sustainable Agriculture. *Sensors* **2025**, *25*, 1398. [CrossRef] [PubMed]
14. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 13289–13298. [CrossRef]
15. Raj, A.; Thakkar, H.K. Comparative Evaluation of Object Detection Models: Transition from Conventional to Deep Learning-Based Approaches. In Proceedings of the 2025 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC), Bhubaneswar, India, 16–18 May 2025; pp. 1–7. [CrossRef]
16. Sangani, M.; Gala, J.; Patil, C.; Smita, S. ResNet Models in Real World Action: A Comparative Study on Tomato Disease Recognition. In *Technologies for Energy, Agriculture, and Healthcare*; CRC Press: Boca Raton, FL, USA, 2024; pp. 536–545. [CrossRef]
17. Galabuzi, C.; Abdullah, H.; Ahmad, N.; Kaidi, H.M. EfficientNet-Based Deep Learning Neural Network for Accurate Plant Disease Detection. In Proceedings of the 2024 5th International Conference on Smart Sensors and Application (ICSSA), Penang, Malaysia, 10–12 September 2024; pp. 1–6. [CrossRef]
18. Rashid, R.; Aslam, W.; Aziz, R.; Aldehim, G. A Modified MobileNetv3 Coupled with Inverted Residual and Channel Attention Mechanisms for Detection of Tomato Leaf Diseases. *IEEE Access* **2025**, *13*, 52683–52696. [CrossRef]
19. Daraghmi, Y.A.; Naser, W.; Daraghmi, E.Y.; Fouchal, H. Drone-Assisted Plant Stress Detection Using Deep Learning: A Comparative Study of YOLOv8, RetinaNet, and Faster R-CNN. *AgriEngineering* **2025**, *7*, 257. [CrossRef]
20. Khan, Z.; Liu, H.; Shen, Y.; Yang, Z.; Yang, F. Optimizing precision agriculture: A real-time detection approach for grape vineyard unhealthy leaves using deep learning improved YOLOv7 with feature extraction capabilities. *Comput. Electron. Agric.* **2025**, *231*, 109969. [CrossRef]
21. Ahmad, B.; Noon, S.K.; Ahmad, T.; Mannan, A.; Ijaz Khan, N.; Ismail, M.; Awan, T. Efficient Real-Time Detection of Plant Leaf Diseases Using YOLOv8 and Raspberry Pi. *VFAST Trans. Softw. Eng.* **2024**, *12*, 250–259. [CrossRef]
22. Gallagher, J.E.; Oughton, E.J. Surveying You Only Look Once (YOLO) Multispectral Object Detection Advancements, Applications, and Challenges. *IEEE Access* **2025**, *13*, 7366–7395. [CrossRef]
23. Yu, Z.; Wang, G.; Zhang, H.; Zhao, K.; Meng, X.; Guo, J.; Geng, M.; Luo, T.; Zhou, K.; He, X. Evaluation of Spray Performance of Swing-Arm Sprayer on Droplet Deposition on Greenhouse Tomatoes. *Agronomy* **2025**, *15*, 2220. [CrossRef]
24. Narimani, M.; Pourreza, A.; Moghimi, A.; Mesgaran, M.; Farajpoor, P.; Jafarbiglu, H. Drone-based multispectral imaging and deep learning for timely detection of branched broomrape in tomato farms. *Proc. SPIE* **2024**, *13053*, 1305304. [CrossRef]
25. Abulizi, A.; Ye, J.; Abudukelimu, H.; Guo, W. DM-YOLO: Improved YOLOv9 model for tomato leaf disease detection. *Front. Plant Sci.* **2025**, *15*, 1473928. [CrossRef] [PubMed]
26. Wang, Y.; He, J. Improved Algorithm for Tomato Disease Detection Based on YOLOv8. In Proceedings of the 2024 IEEE 4th International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, 24–26 May 2024; pp. 500–505. [CrossRef]
27. Arjun, K.; Shine, L.; Deepak, S. Pest Detection and Disease Categorization in Tomato Crops using YOLOv8. In Proceedings of the 2024 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Kothamangalam, India, 16–18 May 2024; pp. 1–6. [CrossRef]
28. Wang, Q.; Yan, N.; Qin, Y.; Zhang, X.; Li, X. BED-YOLO: An Enhanced YOLOv10n-Based Tomato Leaf Disease Detection Algorithm. *Sensors* **2025**, *25*, 2882. [CrossRef] [PubMed]