Research article

https://revistas.unal.edu.co/index.php/refame

Revista Facultad Nacional de Agronomía

Tomato (*Solanum lycopersicum* L.) leaf disease detection using computer vision

UNIVERSIDAD NACIONAL DE COLOMBIA

Detección de enfermedades en las hojas del tomate (*Solanum lycopersicum* L.) mediante visión por computadora

https://doi.org/10.15446/rfnam.v78n3.116493

Sebastian Palacio Betancur[1]* and Freddy Bolaños Martínez[1]

ABSTRACT

Keywords:
Deep learning
Object detection
Plant pathology
PlantVillage dataset
Precision agriculture

Tomato cultivation is a key component of global agriculture, but is significantly threatened by pests and diseases that impact yield and trade. To address this, the present study investigates the application of YOLOv9, a state-of-the-art object detection model, for automated disease detection in tomato leaves. Using a dataset of 4,323 images with 15,135 annotations and a modified PlantVillage dataset, YOLOv9 models were trained and evaluated. Among the evaluated models, YOLOv9e yielded the highest mean average precision (mAP) at 0.964, establishing a benchmark for accuracy. In contrast, the YOLOv9t model provided an optimal balance for practical applications, achieving a competitive mAP of 0.95 with a rapid inference time of 8.8 ms. Furthermore, this work contributes a public version of the PlantVillage dataset with bounding box annotations, providing a valuable resource for object detection research and extending the use of the original classification-focused dataset. The results indicate that YOLOv9 models are effective for real-time and accurate detection of various diseases in complex agricultural settings.

RESUMEN

Palabras clave:
Aprendizaje profundo
Detección de objetos
Fitopatología
PlantVillage dataset
Agricultura de precisión

El cultivo de tomate es un componente clave de la agricultura mundial, pero se ve amenazado significativamente por plagas y enfermedades que impactan el rendimiento y el comercio. Para abordar esto, el presente estudio investiga la aplicación de YOLOv9, un modelo de detección de objetos de última generación, para la detección automatizada de enfermedades en las hojas de tomate. Utilizando un conjunto de datos de 4.323 imágenes con 15.135 anotaciones y un conjunto de datos PlantVillage modificado, se entrenaron y evaluaron los modelos YOLOv9. Entre los modelos evaluados, YOLOv9e arrojó la precisión media promedio (mAP) más alta con 0,964, estableciendo un punto de referencia para la exactitud. En contraste, el modelo YOLOv9t proporcionó un equilibrio óptimo para aplicaciones prácticas, logrando un mAP competitivo de 0,95 con un tiempo de inferencia rápido de 8,8 ms. Además, este trabajo aporta una versión pública del conjunto de datos PlantVillage con anotaciones de cuadros delimitadores, proporcionando un recurso valioso para la investigación en detección de objetos y extendiendo el uso del conjunto de datos original enfocado en la clasificación. Los resultados indican que los modelos YOLOv9 son eficaces para la detección precisa y en tiempo real de diversas enfermedades en entornos agrícolas complejos.

[1]Departamento de Energía Eléctrica y Automática, Facultad de Minas, Universidad Nacional de Colombia Sede Medellín, Colombia. spalaciob@unal.edu.co, fbolanosm@unal.edu.co
*Corresponding author

Received: March 11, 2025; Accepted: June 28, 2025

ISSN 0304-2847 / e-ISSN 2248-7026

The cultivation of tomatoes (*Solanum lycopersicum* L.) represents one of the most significant agricultural activities worldwide, both in terms of production and consumption. According to data from the Food and Agriculture Organization of the United Nations (FAO), approximately 186.1 million tons of tomatoes were produced globally in 2022.

During the same year, it is estimated that around 875,436.86 tons of tomatoes were produced in Colombian territory (FAO 2024), underscoring the importance of this crop in the national agricultural context and its contribution to the country's food security and economy.

However, tomato cultivation, like many other crops, faces significant challenges due to plant pests and diseases. These plant pests are responsible for global losses of crops up to 40% and cause annual trade losses of over 220 billion USD (IPPC Secretariat 2021). The conventional response has been a heavy reliance on chemical inputs, with 4.2 million tons of pesticides used worldwide in 2019 (FAO 2021), a practice that raises significant environmental and public health concerns. In this context, early and accurate disease detection emerges as a key strategy to enable targeted interventions, thereby optimizing the use of agrochemicals and promoting more sustainable agricultural practices.

Traditionally, this task of disease detection has been carried out through visual inspection by agricultural experts. However, this approach presents significant limitations in terms of cost, time, and difficulties in implementation in rural areas where access to experts may be limited.

Computer vision has become a powerful tool for detecting tomato leaf diseases using automatic image techniques. The predominant approach is deep learning, which outperforms traditional machine learning by automatically learning relevant features, without the need for manual feature extraction. This adaptability is especially useful given the complexity and variability of tomato leaf images (Tan et al. 2021).

In the field of computer vision applied for detecting tomato leaf diseases, various techniques have been explored to improve the accuracy and generalization of models. These include the use of hybrid models that combine traditional machine learning with deep learning techniques (Singh et al. 2022), data augmentation through techniques such as Conditional Generative Adversarial Networks (cGAN) (Abbas et al. 2021), the incorporation of pre-trained neural networks such as AlexNet, GoogleNet, DenseNet, SqueezeNet (Abbas et al. 2021; Durmus et al. 2017), and with the use of different versions of You Only Look Once (YOLO) models for plant disease detection (Chairma et al. 2023; Liu and Wang 2020).

It is important to distinguish between classification and detection in this context. While classification focuses on assigning a specific label to an image (i.e., identifying the disease present), detection involves locating and classifying multiple diseases simultaneously in one image, which is crucial for practical real-time applications.

In the specific task of classifying tomato diseases, Mohanty et. al. (2016) used convolutional neural networks (CNN), for the sake of identifying 14 distinct classes of vegetal pathologies, including several which affect tomatoes. Accuracy reached in this case was 99 % under controlled circumstances (Mohanty et al. 2016). The main drawback of this reported solution is its low performance when facing real-world images, and varying conditions such as illumination, background, resolution of the images, and so on. Low computational complexity is also a key issue, since the classification models are intended to be implemented in mobile, real-time, and low consumption platforms. Some modes, such as MobileNet, EfficientNet, and YoloV5-Lite, have been reported in literature for these purposes (Fuentes et al. 2017).

Regarding the detection of tomato diseases, some deep learning models have been proposed. Among the plethora of alternatives reported in literature, some approaches such as Mask R-CNN, PLPNet, and Yolo highlight (Tang et al. 2023). Recent developments in this subject aim to achieve both high accuracy and low delay time, though such objectives seem to be in conflict.

Despite some of the reported advancements, automatic disease detection models for tomato leaves still face several challenges that impact their performance and accuracy. Issues include unwanted elements in images (such as grass or soil), the need for large, high-quality datasets, the complexity and computational demands of deep learning models, and difficulties in detecting multiple or similar diseases. Additional challenges arise from varying field conditions, the integration of drones for image capture, and ensuring reliability in open agricultural environments (Sajitha et al. 2024).

Due to the lack of large datasets for training purposes, transfer learning and self-supervised models have been proposed to reduce the model's dependency to large volume of training data. Such is the case of the work reported in Chai et al. (2024), where self-supervised models are proposed for tomato disease recognition. Another proposed approach implies the use of multispectral images, or the complement of traditional RGB images with thermal signals, in order to improve the detection accuracy (Mahlein 2016).

Given these challenges, the YOLOv9 model family presents a promising solution due to its documented efficiency and accuracy in complex environments. To formally assess this potential in the agricultural context, this study aims to evaluate the performance of different YOLOv9 models for detecting tomato leaf diseases in real-time scenarios.

MATERIALS AND METHODS

Datasets

Dataset for model detection training

The dataset used consists of 4,323 images with 15,135 annotations, including images with simple backgrounds, photos taken of plants, and collages of multiple leaves with different diseases, as shown in Figure 1. This diversity is crucial for training a model capable of functioning in complex, real-world field conditions. This dataset is divided into 9 classes, as presented in Table 1, and includes the necessary bounding boxes for detection.

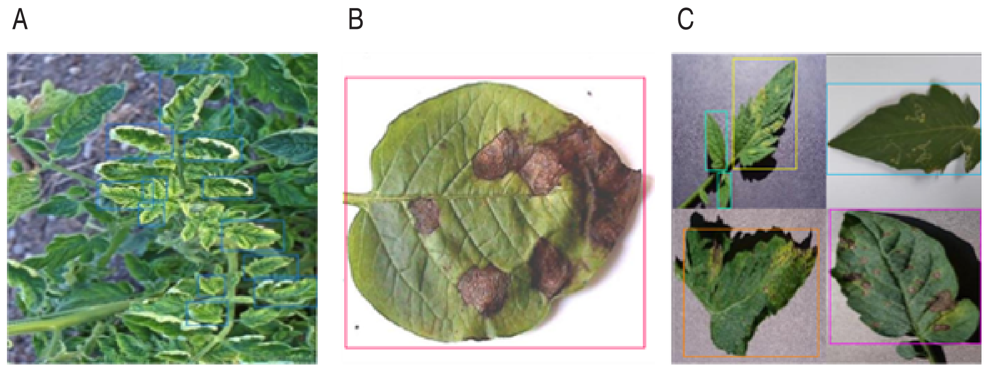

Figure 1. Dataset sample. **A**. Yellow leaf curl virus; **B**. Late Blight and **C**. Multiple classes.

Table 1. Distribution of classes in the dataset.

Class	Number of annotations
Yellow Leaf Curl Virus	2,131
Late Blight	1,886
Leaf Mold	1,878
Mosaic Virus	1,788
Septoria	1,713
Healthy	1,691
Early Blight	1,407
Leaf Miner	1,409
Spider Mites	1,232

This distribution follows a standard machine learning convention, ensuring that the model is trained on a substantial data volume and then evaluated impartially on separate validation and test sets. It is important to note that this dataset is a modified version of a freely available dataset from Roboflow (dyploma 2024) created by a community member. Given its community-sourced origin, guaranteeing the complete precision of every annotation is not feasible. This makes a rigorous evaluation against a standardized benchmark an essential methodological step. For this reason, an additional model validation is performed using the PlantVillage dataset. PlantVillage, being a curated collection of images with single leaves on uniform backgrounds, is less ideal for training a model for complex scenarios but serves as a perfect, unbiased source for validation.

Seventy-five percent of the images were allocated for training, 15% for validation, and 10% for testing.

This distribution follows a standard machine learning convention, ensuring that the model is trained on a substantial data volume and then evaluated impartially on separate validation and test sets. To address class imbalance and enhance model generalization, the "Healthy" class was enriched by augmenting its image count. A structured data augmentation pipeline was applied to the training dataset using the Roboflow platform. This process included converting 15% of the training images to grayscale to reduce color-based bias and improve robustness to variable lighting conditions. Additionally, random crops were applied within bounding boxes with a zoom range of 0–20%, simulating occlusion scenarios and improving localization accuracy for diseases with small lesions, such as Septoria. These augmentations increased the training dataset size from 3,244 to 9,732 images, tripling the available data for model training. Both the enriched and augmented versions of the dataset have been publicly released on Kaggle (Palacio 2024a) to support reproducibility and future research.

PlantVillage dataset for object detection

The PlantVillage is one of the most widely used data sets in machine learning and deep learning research for plant disease classification of images (Sajitha et al. 2024). It contains an extensive collection of images covering 38 different classes, representing various diseases and health states in plants such as apples, cherries, corn, grapes, and tomatoes, among others, with over 54,000 leaf images (Geetharamani and Arun 2019).

However, PlantVillage is mainly designed for classification tasks and does not include the bounding boxes necessary for working with object detection models like YOLO. This limitation has been addressed by some researchers through manual labeling strategies (Nawaz et al. 2022; Mathew and Mahesh 2022), which, while effective, can be extremely labor-intensive and time-consuming.

To address the challenge of labeling the PlantVillage dataset for object detection more efficiently, a semi-automatic labeling pipeline was developed (Figure 2). First, a YOLOv9 model was trained to generate bounding boxes around plant leaves. This initial model processed the entire dataset, producing candidate regions of interest. Next, the dataset's inherent folder structure, where images are organized by class labels (e.g., Tomato_Healthy, Tomato_Late_Blight) was leveraged to automate class assignment. A custom Python script mapped folder names to corresponding annotation labels, assigning classes to the bounding boxes generated by the YOLOv9 model. This approach enabled rapid annotation of 54,000+ images while minimizing manual effort. To ensure accuracy, all annotations were manually reviewed, with <5% of labels requiring correction due to misalignments or edge cases (e.g., overlapping leaves). The final labeled dataset, optimized for object detection tasks, has been publicly released on Kaggle (Palacio 2024b). This resource eliminates the need for labor-intensive manual annotation, providing researchers with a scalable foundation for training and evaluating disease detection models.

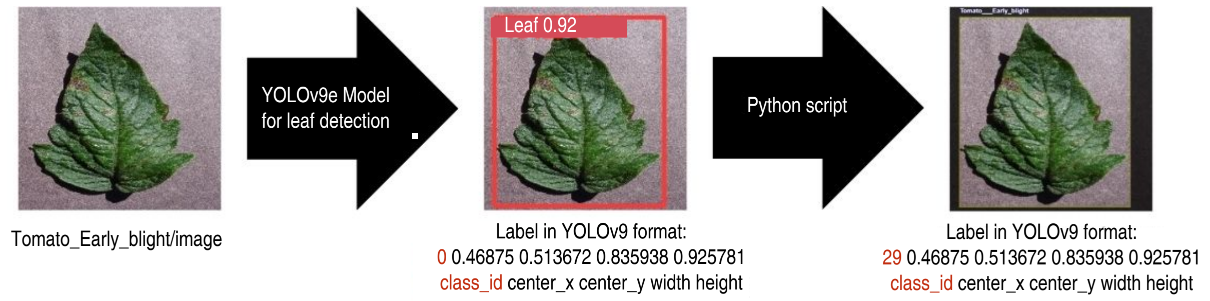

Figure 2. Labeling process for the PlantVillage dataset.

YOLOv9: Detection algorithm

YOLO is a family of deep-learning models designed for real-time object detection in images and videos. Unlike other methods that process images in multiple stages, YOLO processes an entire image in a single pass, dividing it into a grid and assigning bounding boxes and class probabilities to each cell. This architecture allows YOLO to be extremely fast and efficient, making it ideal for applications requiring rapid and precise detection, such as surveillance, autonomous driving, and real-time computer vision.

YOLOv9, the ninth version of this model series, represents a significant advancement in terms of accuracy and efficiency. Introduced in 2024, YOLOv9 incorporates key improvements such as Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN). PGI addresses information loss by integrating a reversible auxiliary branch, enhancing gradient generation across deep layers. GELAN optimizes computational complexity, accuracy, and inference speed by enabling the selection of suitable computational blocks and the use of conventional convolution operators instead of more complex techniques. These innovations have allowed YOLOv9 to reduce the number of parameters by 49% and calculations by 43% compared to its predecessor while improving average precision by 0.6% on the MS COCO dataset. These statistics highlight its ability to deliver precise and efficient real-time detection (Wang et al. 2024).

Table 2, with data sourced from the original YOLOv9 paper (Wang et al. 2024), outlines the specifications for each model variant based on their size and computational needs. Parameters (M) show the number of model parameters, measured in millions, which include the weights and biases the model learns. FLOPs (BFLOPS) indicate the number of floating-point calculations required for one pass through the model, measured in billion FLOPs, reflecting its computational demand.

Table 2. YOLOv9 models.

Model	Parameters (M)	FLOPs (BFLOPS)
YOLOv9t	2.0	7.7
YOLOv9s	7.2	26.7
YOLOv9m	20.1	76.8
YOLOv9c	25.5	102.8
YOLOv9e	58.1	192.5

Performance metrics

The following metrics are used to evaluate the performance of the proposed method, expressed in Equations 1, 2, 3

$$\text{Precision (P)} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{Recall (R)} = \frac{TP}{TP+FN} \quad (2)$$

$$\text{Average precision}(AP) = \int_0^1 P(R)\,dR \quad (3)$$

Intersection over Union (IoU)=(*Area of overlap*)/(*Area of union*)

Where TP is the True Positives, FP is the False Positives, FN is the False Negatives, and C is the Number of Classes.

AP calculates the area under the precision-recall curve, providing a single value that summarizes the model's precision and recall performance across various thresholds (Jocher et al. 2024).

Mean average precision (mAP) extends the concept of AP by averaging the AP values across multiple object classes (Equation 4). This metric is useful in multi-class object detection scenarios to provide a comprehensive evaluation of the model's performance (Jocher et al. 2024).

$$\text{Mean average precision}(mAP) = \frac{1}{C}\sum_{i=1}^{C} AP_i \quad (4)$$

Intersection over Union (IoU): IoU is a measure that quantifies the overlap between a predicted bounding box and a ground truth bounding box. It plays a fundamental role in evaluating the accuracy of object localization (Jocher et al. 2024).

Confusion Matrix: The confusion matrix provides a detailed view of the results by showing the counts of true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs) for each class (Jocher et al. 2024).

Inference Time: Indicates the time taken by the model to process a single image and produce predictions. This metric reflects the model's efficiency and speed in real-time applications.

RESULTS AND DISCUSSION

Training a custom YOLOv9 on a custom dataset

Training an object detection model like YOLOv9 involves teaching the network to recognize and localize objects within images by learning associations between visual features and annotated bounding boxes. This process requires a labeled dataset comprising images annotated with class labels and spatial coordinates (bounding boxes) that define the location of objects. During training, the model iteratively adjusts its parameters to minimize prediction errors, learning to correlate specific visual patterns (e.g., color, texture, shape) with their corresponding classes and spatial positions.

In this work, YOLOv9 models were trained using the code available on the official GitHub repository of the model. The training process was executed using Kaggle Notebooks, which provided access to a free Nvidia Tesla P100 GPU with 16 GB of VRAM. Each model was trained for 120 epochs, using images with 640x640 pixels. All other hyperparameters (e.g., learning rate, batch size, optimizer settings) and training configurations were retained from the repository's default values to ensure alignment with the model's standardized implementation.

Validation with the model's training dataset

The initial validation results were obtained using 15% of the images from the dataset used to train the model, with an image of 640x640 pixels.

Table 3 shows YOLOv9e achieved the highest precision (0.945) and mAP (0.964) due to its deeper network (58.1M parameters) and advanced features like PGI, which retains gradient information for fine-grained feature learning. However, its computational cost (192.5 BFLOPs) results in slower inference (48.4 ms), making it less suitable for edge devices. YOLOv9t sacrifices marginal precision (0.92) and mAP (0.95) for speed (8.8 ms), leveraging a lightweight architecture (2.0M parameters, 7.7 BFLOPs). This trade-off aligns with agricultural needs like drone-based monitoring, where real-time performance is critical.

Table 3. Performance metrics of the models.

Model	Precision	Recall	mAP	Inference Time (ms)
YOLOv9t	0.92	0.867	0.95	8.8
YOLOv9s	0.924	0.884	0.953	11.8
YOLOv9m	0.906	0.91	0.949	25.2
YOLOv9c	0.927	0.893	0.944	35.7
YOLOv9e	0.945	0.904	0.964	48.4

To further analyze the classification performance of the YOLOv9e model on the primary dataset, a confusion matrix was generated, as shown in Figure 3. For object detection tasks, it is crucial to note that this matrix evaluates both classification accuracy and localization precision. A prediction is only considered a True Positive if the class is correct and the Intersection over Union (IoU) between the predicted and ground truth bounding boxes exceeds a set threshold (0.7). The matrix reveals high values along the main diagonal for most classes, such as Leaf Miner (1.00) and Late Blight (0.95), indicating strong performance in both classifying and localizing these diseases. However, the most significant finding is the rate of False Negatives, represented by the background column. For instance, 31% of Yellow Leaf Curl Virus instances and 27% of Healthy leaf instances were categorized as background misses. This type of error occurs for one of two reasons: either the model completely failed to detect an object, or it detected the object but with a bounding box of poor quality ($IoU< 0.7$). In contrast, confusion between the different disease classes was minimal. These results therefore indicate that the primary challenge for the model lies in its detection and localization capabilities, rather than in its ability to differentiate between the pathologies themselves.

Figure 4 shows an example of a YOLOv9e prediction.

Validation with the PlantVillage dataset

For this additional validation, the classes that matched between the model's dataset and the PlantVillage dataset were selected. The classes included are the same as those listed in Table 1, except for Leaf Miner, which is not present in the PlantVillage dataset. A total of 14,627 images were used, each with a size of 256x256 pixels.

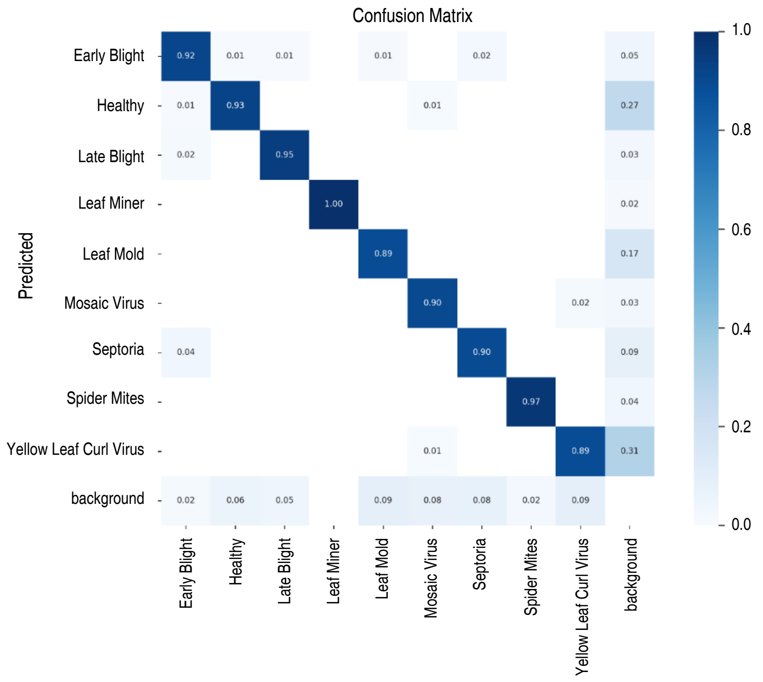

Figure 3. YOLOv9e confusion matrix.

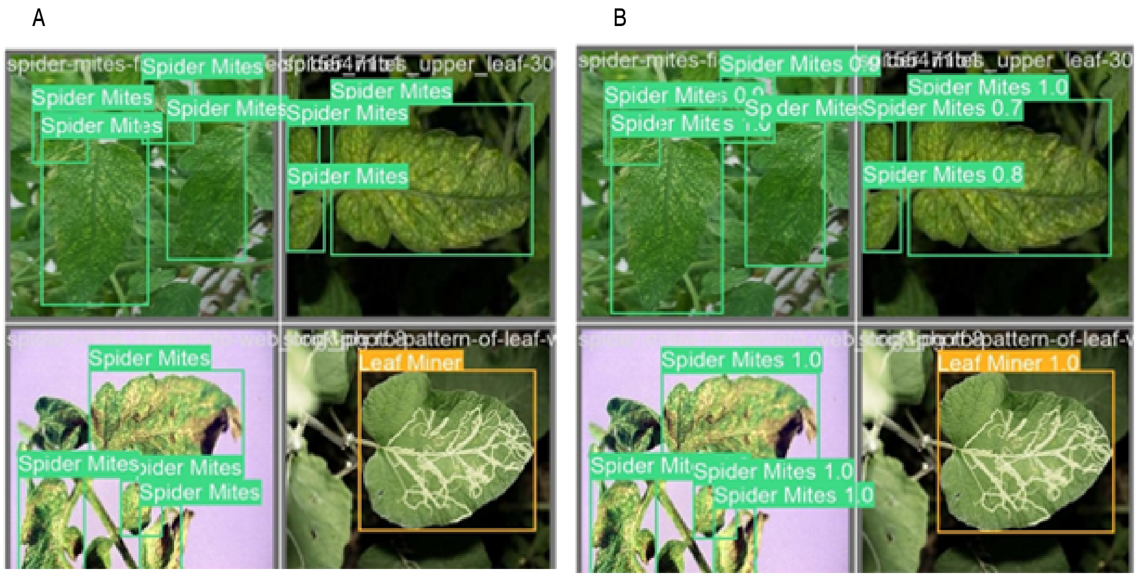

Figure 4. YOLOv9e prediction. **A.** Expected prediction and **B.** Model output.

In Table 4, YOLOv9t demonstrated superior performance on the PlantVillage dataset, achieving the highest precision (0.939), recall (0.933), and fastest inference time (1.4 ms). Conversely, YOLOv9e, despite its higher complexity (58.1 M parameters, PGI, and GELAN layers), underperformed relative to its results on the custom dataset, with precision (0.893), recall (0.911), and mAP (0.958) trailing smaller models.

Table 4. Performance metrics of the models on the PlantVillage dataset.

Model	Precision	Recall	mAP	Inference Time (ms)
YOLOv9t	0.939	0.933	0.965	1.4
YOLOv9s	0.913	0.919	0.952	2.3
YOLOv9m	0.894	0.883	0.942	4.8
YOLOv9c	0.905	0.896	0.955	6.5
YOLOv9e	0.893	0.911	0.958	9.8

This discrepancy could be attributed to the simplicity of PlantVillage images, which feature uniform backgrounds and controlled lighting, potentially reducing the need for YOLOv9e's advanced feature-extraction capabilities. The model's architectural complexity, optimized for diverse real-world conditions (e.g., overlapping leaves, variable lighting), may have led to overfitting on the more heterogeneous training data, plausibly diminishing its adaptability to simpler, structured datasets. Smaller models like YOLOv9t, with fewer parameters (2.0 M) and lower computational demands (7.7 BFLOPs), could be better suited for such scenarios, suggesting that their efficiency aligns with the dataset's reduced complexity.

These findings underscore the importance of aligning model architecture with dataset characteristics for optimal deployment e.g., lightweight models for high-throughput edge devices versus high-precision models for detailed field diagnostics.

The confusion matrix for the YOLOv9t model on the PlantVillage dataset (Figure 5) reveals an exceptionally high level of accuracy. The model achieved perfect or near-perfect classification for several classes, including Healthy (1.00) and Mosaic Virus (1.00), with all other disease classes scoring 0.92 or higher on the main diagonal.

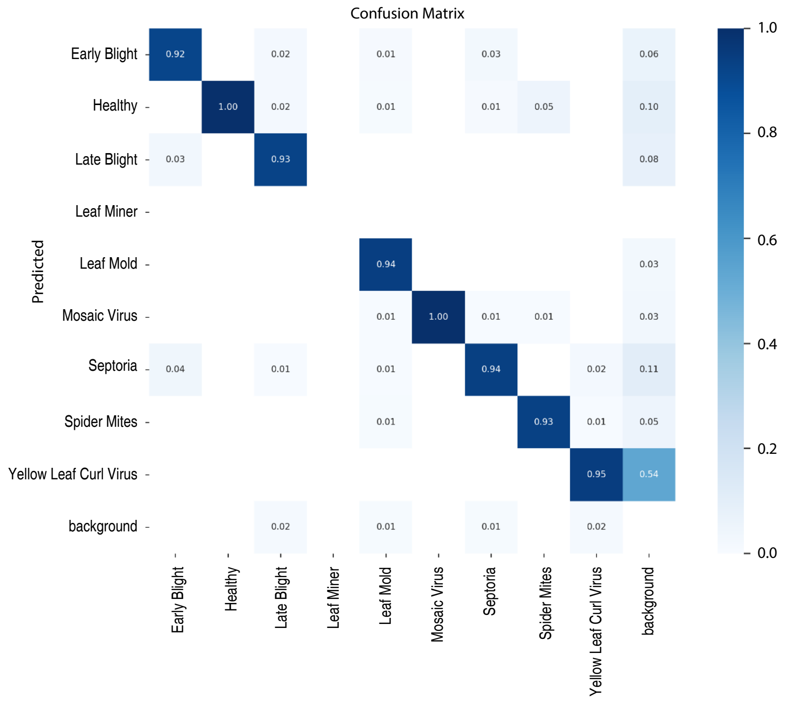

Figure 5. YOLOv9e confusion matrix PlantVillage dataset validation.

Most notably, the False Negative rate, represented by misclassifications into the 'background' class, is reduced across all categories when compared to the model's performance on the more complex primary dataset. This finding strongly supports the previous analysis: the primary source of error encountered in the initial tests was directly related to environmental complexity and challenges in object localization, rather than an inherent limitation in the model's ability to differentiate between pathologies. When tested on clean, uniform images, the model's performance is excellent.

Figure 6 shows an example of a YOLOv9t prediction.

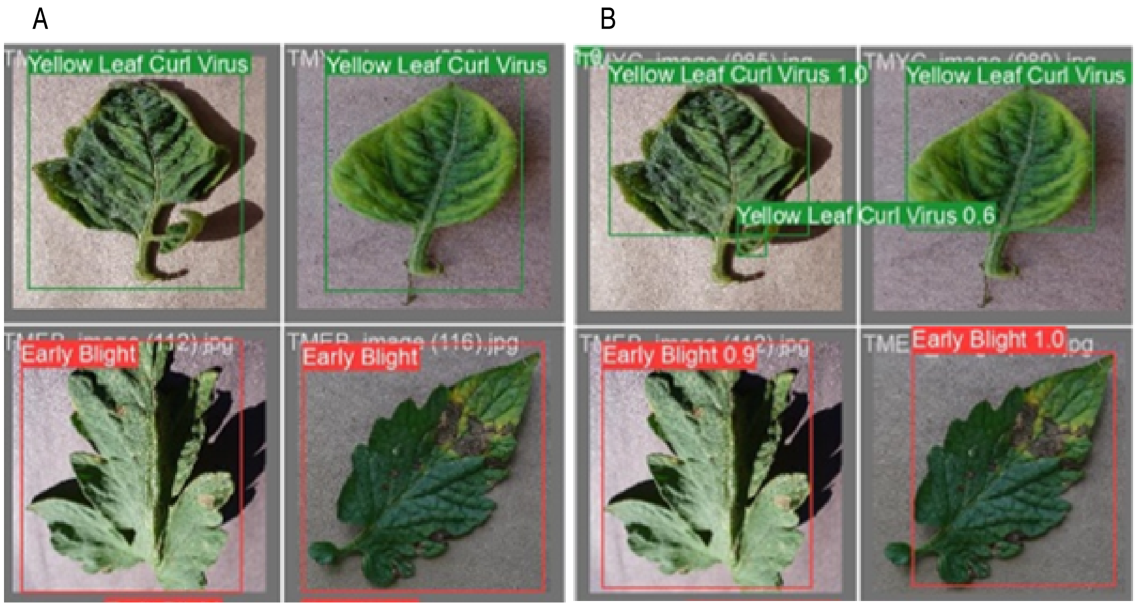

Figure 6. YOLOv9t prediction. **A**. Expected prediction and **B**. Model output.

CONCLUSION

YOLOv9 models exhibit robust performance in tomato leaf disease detection, effectively balancing the dual demands of accuracy and computational efficiency. Among the variants, YOLOv9t emerges as a practical choice for real-time applications, achieving a competitive mean average precision (mAP) of 0.95 with an inference time of 8.8 ms per image. This efficiency—enabled by its lightweight architecture (2.0 M parameters) and reduced FLOPs (7.7 billion)—positions it as ideal for deployment in resource-constrained environments, such as drone-based field monitoring systems requiring rapid, high-throughput processing. In contrast, YOLOv9e prioritizes precision, attaining the highest mAP (0.964) through its advanced Programmable Gradient Information (PGI) and deeper network structure. While its slower inference speed (48.4 ms) may limit its use in time-sensitive scenarios, it remains invaluable for precision-critical tasks, such as greenhouse diagnostics or laboratory analysis, where accuracy is paramount. These findings underscore the importance of model selection based on operational priorities, offering clear guidance for agricultural stakeholders. Notably, even the slowest model (YOLOv9e) operates within real-time thresholds ($\geq$20 FPS), ensuring broad applicability across diverse settings.

A key outcome of this work is the public release of the PlantVillage dataset with bounding box annotations. This initiative adapts the well-known dataset, originally designed for image classification, for a new application in object detection. Providing this resource to the research community removes the need for individual, labor-intensive labeling efforts on this specific dataset and facilitates the training and evaluation of object detection models on a widely recognized set of images. Despite its contributions, this study has certain limitations that should be acknowledged. The training data predominantly featured images from controlled or semi-controlled environments, which may not fully represent the variability of real-world field conditions. Consequently, the model's robustness could be challenged by factors such as motion blur, variable lighting, and partial leaf occlusions not extensively covered in the training set.

These limitations, however, define clear and compelling directions for future research. First, future work should prioritize validating the models on imagery captured directly from field conditions via drones or handheld devices, which is essential for assessing real-world performance. Furthermore, it is recommended to expand the dataset to include other significant plant stressors, such as nutrient deficiencies and pest infestations, to develop a more comprehensive diagnostic tool. A final and logical progression would be the integration of these validated models into edge-computing or IoT platforms to create and test automated, closed-loop disease management systems.

ACKNOWLEDGMENTS

The authors would like to thank the following institutions: Universidad Nacional de Colombia, Medellín Headquarters, and Semillero de Instrumentación, Control y Robótica (SinCRo), for their support in the development of the current work.

CONFLICT OF INTERESTS

The authors declare no potential conflicts of interest.

REFERENCES

Abbas A, Jain S, Gour M and Vankudothu S (2021) Tomato plant disease detection using transfer learning with C-GAN synthetic images. Comput Electron Agric 187. https://doi.org/10.1016/j.compag.2021.106279

Chai AYH, Lee SH, Tay FS et al (2024) Beyond supervision: Harnessing self-supervised learning in unseen plant disease recognition. Neurocomputing 610:128608. https://doi.org/10.1016/j.neucom.2024.128608

Chairma Lakshmi KR, Praveena B, Sahaana G et al (2023) Yolo for Detecting Plant Diseases. In: Proceedings of the 3rd International Conference on Artificial Intelligence and Smart Energy, ICAIS 2023. Institute of Electrical and Electronics Engineers Inc., pp 1029–1034.

Durmus H, Gunes EO and Kirci M (2017) Disease detection on the leaves of the tomato plants by using deep learning. In: 2017 6th International Conference on Agro-Geoinformatics. IEEE, pp 1–5.

dyploma (2024) Tomato Leaf Diseases Dataset. Roboflow Universe.

FAO - Food and Agriculture Organization of the United Nations (2024) FAOSTAT. In: FAOSTAT. https://www.fao.org/faostat/en/#data/QCL

FAO - Food and Agriculture Organization of the United Nations (2021) World Food and Agriculture – Statistical Yearbook 2021. FAO, Rome.

Fuentes A, Yoon S, Kim S and Park D (2017) A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors 17: 2022. https://doi.org/10.3390/s17092022

Geetharamani G and Arun Pandian J (2019) Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Computers & Electrical Engineering Journal 76: 323–338. https://doi.org/10.1016/j.compeleceng.2019.04.011

IPPC Secretariat (2021) International year of plant health – Final report. FAO, Rome.

Jocher G, Munawar MR and Vina A (2024) YOLO Métricas de Rendimiento. YOLO Métricas de rendimiento - Ultralytics YOLOv8 Docs.

Liu J and Wang X (2020) Tomato Diseases and pests detection based on improved Yolo V3 Convolutional Neural Network. Front Plant Sci 11: 521544. https://doi.org/10.3389/fpls.2020.00898

Mahlein A-K (2016) Plant disease detection by imaging sensors – parallels and specific demands for precision agriculture and plant phenotyping. Plant Diseases 100: 241–251. https://doi.org/10.1094/PDIS-03-15-0340-FE

Mathew MP and Mahesh TY (2022) Leaf-based disease detection in bell pepper plant using YOLO v5. Signal Image Video Process 16: 841–847. https://doi.org/10.1007/s11760-021-02024-y

Mohanty SP, Hughes DP and Salathé M (2016) Using deep learning for image-based plant disease detection. Front Plant Sci 7. https://doi.org/10.3389/fpls.2016.01419

Nawaz M, Nazir T, Javed A et al (2022) A robust deep learning approach for tomato plant leaf disease localization and classification. Sci Rep 12. https://doi.org/10.1038/s41598-022-21498-5

Palacio S (2024a) Tomato leaf diseases dataset for object detection. https://www.kaggle.com/dsv/9270440

Palacio S (2024b) PlantVillage for object detection YOLO. https://www.kaggle.com/ds/5363572

Sajitha P, Diana Andrushia A, Anand N and Naser MZ (2024) A review on machine learning and deep learning image-based plant disease classification for industrial farming systems. J Ind Inf Integr 100572. https://doi.org/10.1016/j.jii.2024.100572

Singh A, Sreenivasu S, Mahalaxmi U et al (2022) Hybrid Feature-Based disease detection in plant leaf using convolutional neural network, bayesian optimized SVM, and random forest classifier. J Food Qual 2022: 16. https://doi.org/10.1155/2022/2845320

Tan L, Lu J and Jiang H (2021) Tomato leaf diseases classification based on leaf images: A comparison between classical machine learning and deep learning methods. AgriEngineering 3: 542–558. https://doi.org/10.3390/agriengineering3030035

Tang Z, He X, Zhou G et al (2023) A precise image-based tomato leaf disease detection approach using PLPNet. Plant Phenomics 5:0042. https://doi.org/10.34133/plantphenomics.0042

Wang CY, Yeh I-H and Liao HYM (2024) YOLOv9: Learning what you want to learn using programmable gradient information.