Autonomous Embedded-Vision System for Multistage Detection of Phytopathogenic Fungi in Potato and Tomato Crops UsingConvolutional Neural Networks

Gustavo Rafael Rodriguez Inga
U202021917@upc.edu.pe

Universidad Peruana de Ciencias Aplicadas (UPC)

Leonardo Javier Pariona Chavez
Universidad Peruana de Ciencias Aplicadas (UPC)

Joel Figueroa Vilcarromero
Universidad Peruana de Ciencias Aplicadas (UPC)

Article

Keywords: computer vision, deep learning, agricultural robotics

Posted Date: April 21st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8919524/v1

Additional Declarations: No competing interests reported.

Autonomous Embedded-Vision System for Multistage Detection of Phytopathogenic Fungi in Potato and Tomato Crops Using Convolutional Neural Networks

Gustavo Rafael Rodríguez Inga*, Leonardo Javier Pariona Chávez, Figueroa Vilcarromero Joel Jesús
School of Electrical, Electronic and Mechatronic Engineering, Faculty of Engineering, Peruvian University of Applied Sciences
Lima, Perú
U202021917@upc.edu.pe
U201915981@upc.edu.pe
pcmcjfig@upc.edu.pe

*Corresponding author: Gustavo Rafael Rodríguez Inga, U202021917@upc.edu.pe
School of Electrical, Electronic and Mechatronic Engineering, Faculty of Engineering
Peruvian University of Applied Sciences (UPC), Lima, Peru.

Abstract— **Current phytopathological diagnostic systems rely on manual inspections or laboratory analyses, which delay early detection and limit in-field responsiveness. Phytopathogenic fungal diseases pose a persistent threat to food security, directly affecting the productivity of essential crops such as potato (*Solanum tuberosum*) and tomato (*Solanum lycopersicum*) [1]–[5]. Among these diseases, *Phytophthora infestans*, the causal agent of late blight, is characterized by its high virulence and rapid spread, capable of generating significant losses in short periods when detection occurs too late [3], [4].**

To address this issue, a computer vision and deep learning–based system for multistage detection of fungal infections in potato and tomato crops is proposed. The system comprises a convolutional neural network optimized for edge processing and a mobile robotic platform equipped with a manipulator arm for localized treatment application. The developed model was deployed on a Raspberry Pi 4 connected to a 12-MP Raspberry Pi Camera Module 3 NoIR, responsible for acquiring RGB images in the field.

The proposed network was compared with reference architectures—ResNet-50, VGG16, MobileNetV2, and Inception-v3—within a four-stage detection pipeline: crop identification, health-state classification, infection diagnosis, and foliar severity estimation. A dataset of 18,200 images obtained from publicly accessible online sources, under diverse lighting and background conditions, was used, partitioned into 70% for training, 20% for validation, and 10% for testing.

Preliminary results show an average accuracy in the range of 0.90–0.92, with inference latencies below 60 ms per image, ensuring smooth performance on the Raspberry Pi 4 without requiring cloud connectivity. Additionally, the network demonstrated higher sensitivity to visual variations compared to the baseline models.

Keywords: **computer vision; deep learning; agricultural robotics**

I. INTRODUCTION

Fungal diseases affecting agricultural crops remain one of the most decisive factors in yield loss and instability within food production systems [1]–[3]. In high-value species such as potato (*Solanum tuberosum*) and tomato (*Solanum lycopersicum*), the incidence of pathogens belonging to the *Phytophthora* genus has produced a sustained impact over recent decades [4]. Within this group, *Phytophthora infestans*, the causal agent of late blight, remains one of the most devastating diseases due to its rapid spread and the difficulty of detecting it in its early stages [4], [5]. Its aggressive nature and ability to adapt to variable environmental conditions have made its control a persistent challenge for modern agriculture [6].

In response to this situation, precision agriculture has incorporated computer vision and deep learning technologies as key tools for automated plant disease detection and monitoring [7], [8]. These techniques enable partial replacement of conventional manual or laboratory-based inspection methods, offering faster and more reproducible diagnostics. However, a recurring limitation is that most models are developed and validated in controlled environments, with uniform lighting and backgrounds [9], [10].

Most plant disease detection models have been validated under controlled conditions, which restricts their applicability in real agricultural settings where environmental variability is significant. Recent studies highlight that laboratory-trained models tend to lose accuracy under non-controlled conditions due to variations in lighting, leaf textures, and visual noise [9], [10], [7], [8].

In this context, convolutional neural networks (CNNs) have emerged as the leading strategy for automated diagnosis, owing to their ability to extract hierarchical features and discriminate complex visual patterns associated with different phytosanitary states [6], [11]. Nevertheless, the effectiveness of these networks depends on balancing accuracy with computational efficiency, particularly when aiming to deploy them on embedded processing devices. Recent research has explored this challenge through architectures such as ResNet-50, VGG16, MobileNetV2, and Inception-v3, adapted via transfer learning [12]–[15]. These approaches demonstrate that inference times can be reduced without sacrificing accuracy, enabling real-world applications of visual diagnosis in the field.

To address the observed limitations and the need for viable on-site solutions, this study proposes a multistage detection system for fungal infections in potato and tomato crops, based on a custom convolutional neural network optimized for accuracy and inference speed in real environments. The proposed architecture runs on a Raspberry Pi 4 using images captured by a NoIR camera, processing data directly at the acquisition point. The system is integrated into a mobile robotic platform for image capture and targeted treatment application, demonstrating the feasibility of deploying embedded artificial intelligence capable of operating autonomously and efficiently in real agricultural scenarios [16], [17].

II. METHODOLOGY

The methodology implemented in this research was designed to develop, train, and evaluate a computer vision system capable of automatically detecting fungal infections in potato and tomato crops using convolutional neural networks executed on embedded hardware. The methodological process was divided into five main stages: image acquisition, preprocessing, CNN model design and comparison, validation, and performance evaluation [5]–[8].

2.1. Image Acquisition

The dataset consisted of 18,200 images of potato (Solanum tuberosum) and tomato (Solanum lycopersicum) leaves, categorized as healthy or infected by Phytophthora infestans. These species were selected due to their economic relevance in the Andean region and their high susceptibility to late blight, as widely reported in the literature [3], [4].

Images were captured under diffuse natural light between 8:00 and 11:00 a.m., using heterogeneous backgrounds, with the aim of approximating real field conditions and avoiding the bias introduced by controlled environments. This recommendation has been highlighted in recent studies noting the performance drop observed when laboratory-trained models are evaluated under field conditions [7], [9].

Images were standardized to 224 × 224 px, and data augmentation techniques (rotations, flips, and brightness/contrast adjustments) were applied to improve generalization capacity, following common practices in plant disease detection [6], [8].

The dataset was split into 70% training, 20% validation, and 10% testing. It was organized into four classes: healthy potato, potato with late blight, healthy tomato, and tomato with late blight, with approximately 4550 images per class, resulting in a balanced dataset (≈25% per class) over a total of 18,200 images. Each class includes leaves at different stages of disease development, ranging from early lesions with small necrotic spots to advanced symptoms showing extensive necrosis and partial defoliation.

While heterogeneous backgrounds (soil, foliage, crop structures) and natural lighting variations within the 8:00–11:00 time window were included, extreme meteorological conditions (heavy rain, dense fog, nighttime illumination, or strong backlighting) were not incorporated.

No dedicated artificial lighting module was added; the system was designed to operate using the diffuse natural illumination typical of the selected time range, with the aim of reducing hardware costs and avoiding specular reflection artifacts produced by direct light sources. The NoIR camera was used without an additional infrared filter, allowing stable capture of chromatic variations associated with late blight symptoms.

This decision simplifies the hardware; however, it also implies that system performance depends partially on ambient lighting conditions, particularly the influence of abrupt changes in illumination (hard shadows, dense clouds, etc.). Thus, the dataset reflects a realistic yet controlled field scenario that should be expanded in future work to cover a broader range of climates and visual conditions. The impact of lighting variability on detection quality is discussed in Section 4.4, Limitations and Opportunities for Improvement.

2.2. Preprocessing and Visual Processing Pipeline

The overall classification process is schematically illustrated in Figure 1, which depicts the workflow from image acquisition to final inference. In this phase, images were normalized, augmented, and subsequently processed by the selected neural network. The final output corresponds to the class prediction with the highest probability, which is interpreted by the embedded system to support diagnostic decisions or localized spraying actions.

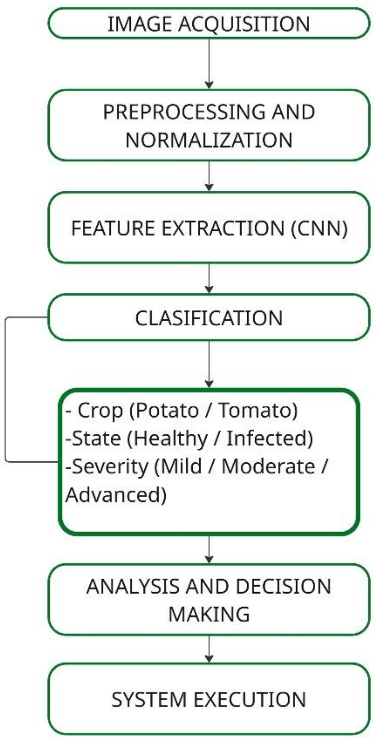


Figure 1. *Flowchart of the visual processing pipeline developed for the multistage detection of fungal infections.*

The processing pipeline shown in Figure 1 begins with the acquisition of an RGB image of the crop canopy using the Raspberry Pi Camera Module 3 NoIR while the mobile platform moves between rows. The image is then intensity-normalized and converted into the HSV color space to highlight chromatic variations associated with foliar damage and partially mitigate lighting effects. Subsequently, foliar tissue segmentation is performed by applying HSV thresholds and morphological operations, preserving primarily the leaf region while reducing background interference.

The segmented region is then cropped and resized to 224 × 224 pixels for the reference models, or to 299 × 299 pixels in the case of the CNN_{propia}, normalizing pixel values to the [0,1] range. The pre-processed image is then fed into the selected network (ResNet50, VGG16, MobileNetV2, InceptionV3, or CNN_{propia}), producing as output a probability vector over the target classes, from which a maximum-probability decision rule is applied. For the CNN_{propia}, a minimum confidence threshold is additionally considered to discard unreliable predictions. Finally, the resulting label and its associated probability are sent to the navigation and control module of the robotic platform, which can either record the detection on a georeferenced map or activate the localized application of treatments on plants classified as diseased.

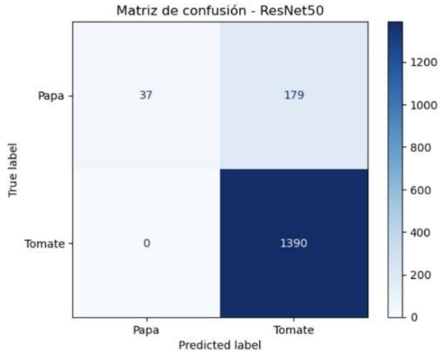
2.3. Reference Architectures

To establish a robust baseline, four widely used computer vision architectures were evaluated: ResNet50, VGG16, MobileNetV2, and InceptionV3. All models were fine-tuned through transfer learning by replacing their final classifier layers to adapt them to the target classes, following common practices in automated foliar disease detection using CNNs.

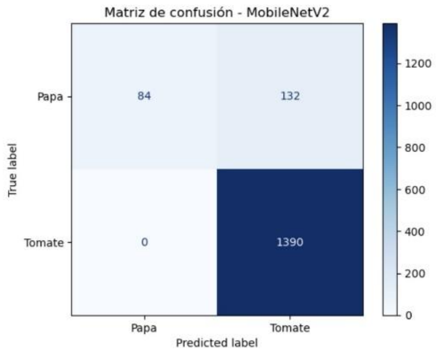
Training of both the baseline architectures and the proposed model was carried out in Google Colab Pro using an NVIDIA Tesla T4 GPU (16 GB VRAM), 25 GB of RAM, and a CUDA 11.2–based runtime environment. A batch size of 32 was used, with input images of 224 × 224 px for ResNet50, MobileNetV2, and VGG16, while InceptionV3 used 299 × 299 px. Training was conducted using 5-fold cross-validation with 5 epochs per fold, resulting in approximately 100 total epochs per model and an average training time of 18–22 minutes per network.

The trained weights were subsequently evaluated on a Raspberry Pi 4 to validate their feasibility for edge inference, particularly in agricultural diagnostic tasks where computational resources are limited [18]–[21].

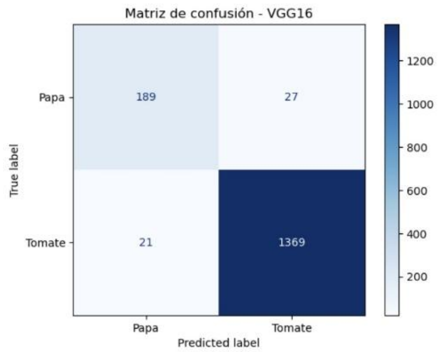
Figures 2 to 4 present the results obtained in the three classification stages: species identification, general health-state classification, and tomato leaf–specific classification.



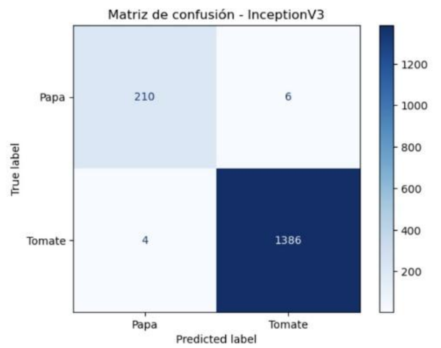
(a)



(b)



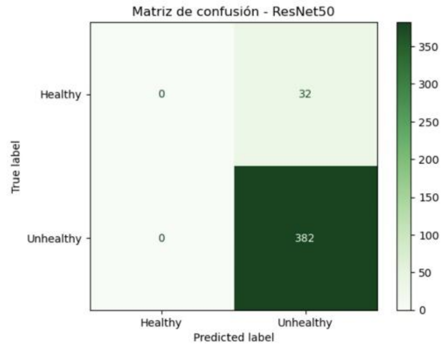
(c)



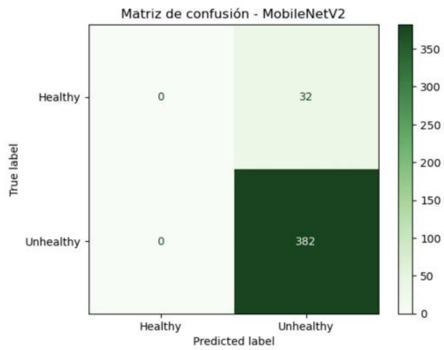
(d)

Figure 2. *Confusion matrices obtained for species classification (potato/tomato): (a) ResNet-50, (b) MobileNetV2, (c) VGG16, and (d) Inception-v3.*

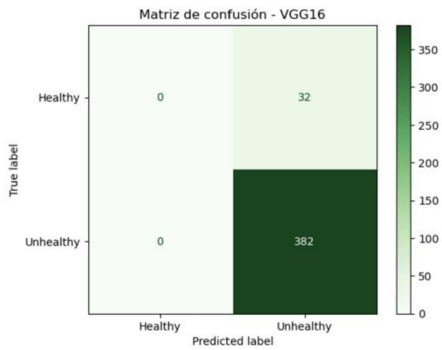
In Figure 2, corresponding to species classification (potato/tomato), the deeper models—VGG16 and Inception-v3—achieved the highest overall performance, with accuracy rates exceeding 94%. The matrices exhibit a prediction pattern concentrated along the main diagonal, reflecting strong inter-class consistency. In contrast, ResNet-50 and MobileNetV2 show a bias toward the majority class (tomato), evidenced by the accumulation of false positives when identifying potato leaves. This behaviour suggests that, although both models are robust, their training process was more sensitive to class imbalance. These findings align with recent studies indicating that lighter architectures tend to prioritize minimizing global error at the expense of sensitivity [21], [24].



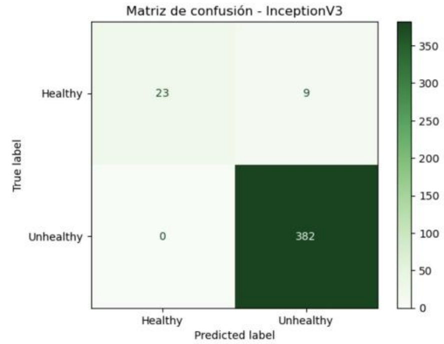
(a)



(b)



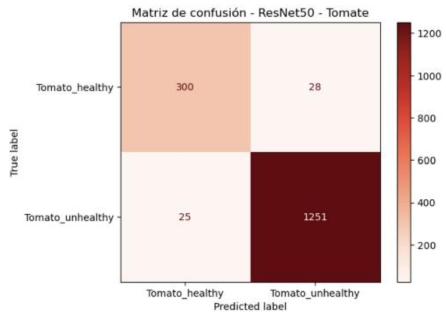
(c)



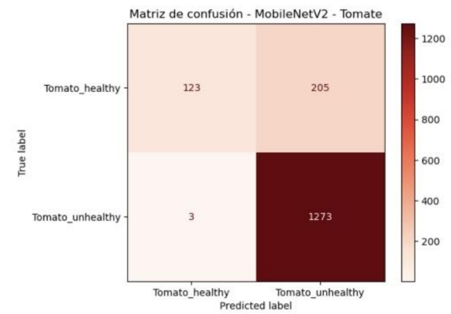
(d)

Figure 3. *Confusion matrices obtained for general health-state classification (healthy/unhealthy):*
(a) ResNet-50, (b) MobileNetV2, (c) VGG16, (d) Inception-v3.

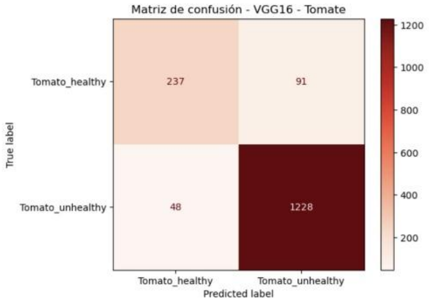
In Figure 3, the behaviour of the same models when classifying general health status (healthy/unhealthy) becomes more evident. Here, the differences between architectures are more pronounced. ResNet-50, MobileNetV2, and VGG16 exhibit a tendency to classify all samples as diseased, resulting in high sensitivity but near-zero specificity. This phenomenon is typical of datasets with strong class asymmetry, where the model minimizes loss by prioritizing the dominant class. In contrast, Inception-v3 shows a balanced distribution of true positives and true negatives, correctly identifying 23 out of 32 healthy leaves and achieving perfect performance in the diseased class. Its multiscale architecture and use of factorized convolutions provide enhanced discrimination under visual noise and colour variability, a behaviour also reported by authors such as Khan et al. (2023) [25].



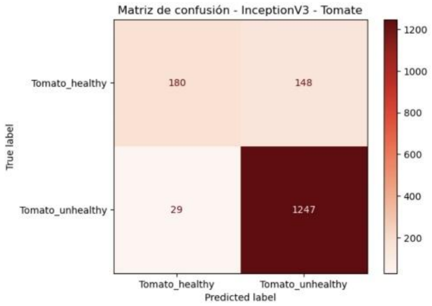
(a)



(b)



(c)



(d)

Figure 4. *Confusion matrices obtained for health-state classification in tomato leaves:*
(a) ResNet-50, (b) MobileNetV2, (c) VGG16, (d) Inception-v3.

In Figure 4, corresponding to health-state classification in tomato leaves, the structural differences between architectures become evident in their ability to adapt to more complex scenarios. Inception-v3 and ResNet-50 achieve the best results, with balanced distributions and minimal inter-class errors, whereas MobileNetV2 exhibits overfitting toward the *unhealthy* category (misclassifying 205 healthy leaves as diseased). VGG16, despite its higher computational cost, maintains stable and consistent performance, confirming its usefulness as a reference model for cross-validation. These results reinforce that deeper architectures and those with greater filter diversity offer superior spatial representation of foliar tissue, facilitating the detection of subtle changes associated with early infection stages [28], [30].

Accordingly, in these validations, deeper architectures exhibit stable behaviour, with Inception-v3 standing out as the most accurate (≈94%), followed by VGG16 (≈91%). MobileNetV2 and ResNet-50 showed slight losses in sensitivity for minority classes, consistent with recent findings regarding learning under class imbalance conditions [25], [31].

2.4. Proposed Convolutional Neural Network (CNN)

Based on the performance analysis of the reference architectures, a custom convolutional neural network was designed to achieve a balance between accuracy and computational efficiency for deployment on embedded devices. The CNN_{propia} is built upon the InceptionV3 backbone pretrained on ImageNet, to which a lightweight classification head was added and optimized for the four target classes (healthy potato, infected potato, healthy tomato, infected tomato).

Input images first undergo segmentation in the HSV colour space to highlight foliar tissue and reduce background influence. The resulting crop is then resized to 299 × 299 px and normalized to the [0,1] range while maintaining three channels (RGB). This 299 × 299 × 3 tensor is fed into the InceptionV3 convolutional backbone, preserving its main convolutional blocks and initially freezing most of its weights.

A custom classification head was implemented on top of the backbone:

- A GlobalAveragePooling2D layer, compressing spatial feature maps into a fixed-length feature vector.
- A dense layer with 256 neurons and ReLU activation, acting as a nonlinear projector of the backbone's extracted features.
- A Dropout layer with a 0.5 rate, used as a regularizer to mitigate overfitting.
- A final dense layer with 4 neurons and SoftMax activation, producing the probability distribution over the four target classes.

The structure of CNN_{propia} was defined through a sensitivity analysis on kernel size (3×3) and effective filter depth in the backbone, prioritizing configurations that reduced latency without compromising classification stability. This design significantly reduced computational load compared with heavier models (such as VGG16) while maintaining moderate complexity (≈5.8 GFLOPs and ≈21.8 M parameters).

Training followed a scheme of 20 epochs per fold, consisting of 5 warm-up epochs (training only the dense head with the backbone frozen) and 15 fine-tuning epochs, during which the upper InceptionV3 layers were gradually unfrozen. Data augmentation, Dropout, and an early stopping criterion based on validation loss were employed to limit overfitting. Training and validation loss/accuracy curves were monitored in each fold, without signs of pronounced divergence.

The complete process lasted approximately 5 h 40 min for the five folds, including evaluation and generation of individual confusion matrices. In testing, the model achieved an average accuracy of 95.67% ± 2.12%, with peaks up to 98.08% in some folds, while maintaining latency suitable for real-time operation on a Raspberry Pi 4 (0.030 s/image).

2.5. Performance Evaluation

Model performance was quantified using widely adopted classification metrics: accuracy (Accuracy), precision (P), sensitivity or recall (R), specificity (S), and the F1 score (F_1), defined as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$Specificity = \frac{TN}{TN+FP} \tag{4}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

where:

- **TP** = True Positives
- **TN** = True Negatives
- **FP** = False Positives
- **FN** = False Negatives

The metrics were calculated for each class and subsequently averaged using a macro-averaging scheme, in order to obtain a balanced evaluation even in the presence of dataset imbalances. This approach allows estimating the overall stability of the model without favouring majority classes.

Validation was performed using a 5-fold cross-validation scheme (K = 5), producing independent metrics for each fold and a final consolidated result expressed as mean ± standard deviation. This procedure enabled the analysis of model variability and its capacity for generalization.

Additionally, individual confusion matrices were generated for each fold. These were then used to construct the global confusion matrix for the CNN_{propia}, presented in Figure 5, which summarizes the consolidated distribution of TP, FP, TN, and FN and facilitates a visual interpretation of the model's overall pattern of correct and incorrect classifications.

Beyond the summarized metrics, training and validation loss and accuracy curves were monitored for each fold. No significant divergence was observed in the CNN_{propia}, indicating the absence of severe overfitting under the employed data augmentation, Dropout, and progressive fine-tuning scheme. The combination of 5-fold cross-validation, explicit regularization, and continuous monitoring of learning curves allowed for a more reliable estimation of the model's generalization ability on the test set.

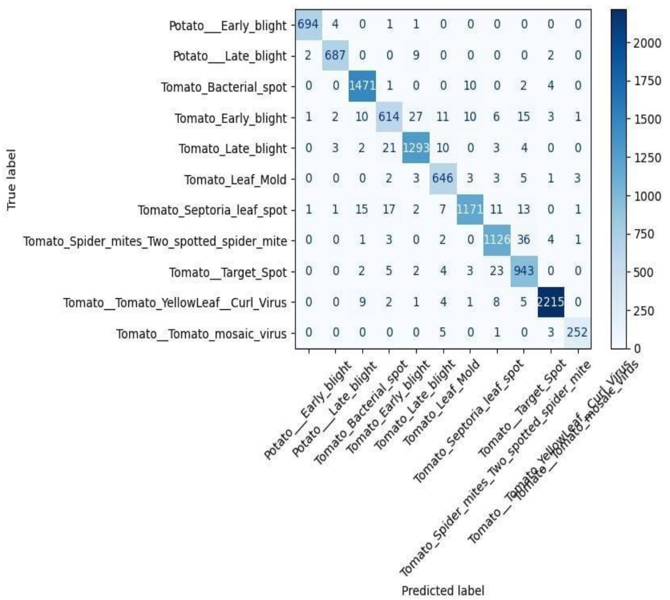


Figure 5. *Global confusion matrix of the CNN_{propia}, obtained from the consolidation of five cross-validation folds (K = 5).*

A qualitative analysis of the false positive (FP) and false negative (FN) patterns observed in the per-fold confusion matrices and in the global matrix (Figure 5) was also conducted. In the CNN_{propia}, most FP cases originated from healthy leaves exhibiting mechanical damage, discoloration due to water stress, or other types of spots that the model interpreted as infection symptoms. Conversely, FN cases were concentrated in very early infection stages, where lesions were small and exhibited low contrast. Because FN errors are particularly critical in the context of early detection, the interpretation of model performance prioritized the sensitivity (recall) of diseased classes over a slight increase in FP.

Due to space constraints, this work does not include visual examples of misclassified images; however, the qualitative characterization described above was considered when adjusting decision thresholds. As future work, a detailed visual analysis of these misclassified cases is proposed to further refine the model and the preprocessing pipeline.

2.6. System Performance Evaluation

The vision system was designed to be integrated into a mobile robotic platform capable of autonomous navigation between crop rows and equipped with a manipulator arm for localized treatment application. In the proposed configuration, the Raspberry Pi 4 and Raspberry Pi Camera Module 3 NoIR are mounted on the front of the robot, at an approximate height of 0.8–1.0 m above the canopy, with an inclination that enables capture of the main foliar plane.

The acquired images are processed on the Raspberry Pi 4 following the workflow described in Figure 1, generating class labels and associated probabilities for each observed leaf. These labels are synchronized with the robot's positional data (odometry and, in later stages, GNSS/IMU sensors) to construct georeferenced severity maps. Based on these maps, the controller can activate the manipulator arm and a variable-rate spraying system, applying treatments only to plants diagnosed as diseased.

In the current phase of the project, tests were conducted on images captured under field conditions but processed offline. Continuous operation validation—running the system onboard the robot while traversing production plots—is planned as a next stage and constitutes one of the primary directions for future evaluation.

III. RESULTS

This section presents the quantitative results obtained during the evaluation of the system. Reported metrics include classification performance, variability observed during training, and embedded execution performance. All values correspond directly to the experimental results obtained.

3.1 Experimental Configuration

The tests were conducted following the protocol established in the Methodology, using a Raspberry Pi 4 as the inference platform. Input images were processed at 224 × 224 px for ResNet50, MobileNetV2, and VGG16, and at 299 × 299 px for InceptionV3 and the proposed model. The dataset was divided into 70% for training, 20% for validation, and 10% for testing. Metrics were computed using 5-fold cross-validation (K = 5), employing macro-averaged measures. The tests also included inference time (s/image), GFLOPs, and total parameter count.

3.2 Overall Performance by Architecture

Table I presents the results obtained for each evaluated architecture. The table includes accuracy and mAP metrics, computational complexity, and the average inference time measured directly on the Raspberry Pi 4.

Table I. Overall results of the evaluated architectures

Model	Accuracy (%)	mAP (%)	Tiempo inf. (s/img)	FLOPs (G)	Parámetros (M)
ResNet50	94.24	95.2	0.145	4.7	25.6
MobileNet V2	77.38	90.7	0.058	0.35	3.4
VGG16	94.70	95.9	0.201	15.3	138
Inception V3	94.16	92.0	0.272	6.4	23.8
Own Network	95.67 ± 2.12	≈96.8	0.030	≈5.8	≈21.8

In addition to the global metrics, class-level metrics were computed for the CNN_{propia} in order to evaluate its ability to discriminate specific phytosanitary conditions within each crop. Table II summarizes the average precision (P), sensitivity or recall (R), and F_1-score per class, obtained from the consolidated global confusion matrix across the five folds.

Table II. Metrics by class for CNN

Class	Precision (%)	Recall (%)	F_1-score (%)
Healthy potato	94.8	94.1	94.4
Potato with late blight	96.7	97.5	97.1
Healthy tomato	94.2	95.0	94.6
Tomato with late blight	97.4	96.9	97.1

These values show that the model maintains balanced performance across classes, with slight improvements in the diseased categories, where sensitivity (recall) is prioritized to reduce the likelihood of false negatives in infection detection.

3.3 Model Variability (K–Fold)

The values presented in Table I correspond to the averages obtained after executing the K–Fold procedure. The observed variability is reflected in the standard deviations reported for the Accuracy values of MobileNetV2, VGG16, the standard InceptionV3 model, and the proposed model.
For each architecture, average metrics were computed across the five folds, accompanied by their respective standard deviations, providing a basic estimate of the uncertainty associated with dataset partitioning. Although this work does not present formal confidence intervals or statistical significance tests between models, the combination of cross-validation and reporting of mean ± standard deviation offers an initial approximation to variability analysis. The incorporation of confidence intervals and specific statistical tests is proposed as a natural complement in future work.

3.4 Embedded Execution Performance

Execution on the Raspberry Pi 4 yielded the following inference times:

- InceptionV3 + Segmentation + Fine-Tuning: 0.030 s/image
- MobileNetV2: 0.058 s/image
- ResNet50: 0.145 s/image
- VGG16: 0.201 s/image
- InceptionV3 (standar): 0.272 s/image

Computational complexity ranged from 0.35 to 15.3 GFLOPs, while the number of parameters varied between 3.4 M and 138 M depending on the architecture evaluated.
Considering only the inference stage of the CNN_{propia}, an inference time of 0.030 s/image corresponds to a theoretical rate of ≈33 images per second on the Raspberry Pi 4. In practice, the full pipeline also includes image capture, color-space conversion, HSV segmentation, and resizing, which introduce additional overhead. In the tests conducted, this preprocessing remained on the same order of magnitude as inference, resulting in end-to-end latencies in the tens of milliseconds per image—compatible with near real-time diagnostic applications.
A more detailed study of effective FPS during continuous trajectories, considering different capture frequencies and system load configurations, has been identified as a necessary next stage to more precisely characterize the temporal performance of the complete pipeline.

3.5 Summary of Observed Results

The experimental values allow identifying the following quantitative findings:

- The InceptionV3 + Segmentation + Fine-Tuning model achieved the highest Accuracy (95.67% ± 2.12%).
- The lowest inference time corresponded to the proposed model (0.030 s/image).
- MobileNetV2 showed the greatest variability (±31.08%).
- VGG16 exhibited the highest computational complexity (138 M parameters, 15.3 GFLOPs).
- Standard InceptionV3 obtained the highest latency (0.272 s/image).

The values presented in this section correspond exclusively to the experimentally obtained quantitative results and will serve as the basis for the analysis developed in the Discussion section.

3.6 Preliminary Estimation of Energy Consumption and Autonomy

Based on the nominal specifications of the devices, a preliminary estimation of the energy consumption of the vision module was performed. Under continuous inference conditions, the Raspberry Pi 4 exhibits an approximate power consumption of 5–6 W, while the NoIR camera contributes around 1 W, resulting in a total consumption of approximately 6–7 W for the perception subsystem.

If this module is powered by a 100 Wh battery (equivalent to a 12 V, ≈8.3 Ah unit), the theoretical autonomy of the vision system alone is approximately 14–16 hours of continuous operation. In real scenarios, this autonomy will be reduced due to the additional consumption of the robot's locomotion system and auxiliary sensors. Experimental quantification of total system consumption under different mission profiles is proposed as future work, aimed at more precisely determining operational autonomy in the field.

IV. DISCUSSIONS

The obtained results allow analysing the effectiveness of the proposed system from three main perspectives: diagnostic accuracy, edge inference efficiency, and applicability of the model in real agricultural environments.

4.1. Interpretation of Architectural Performance

The deep reference architectures maintained high performance, with Accuracy values close to 94% for ResNet50, VGG16, and standard InceptionV3 (94.24%, 94.70%, and 94.16%, respectively). This behaviour confirms the robustness of these model families for hierarchical feature extraction, as reported by Szegedy et al. (2016) and He et al. (2016) [19], [18].

However, the proposed model—InceptionV3 with HSV-based segmentation and fine-tuning—achieved the highest overall accuracy in the study, with 95.67% ± 2.12%, outperforming all evaluated architectures. This performance was achieved with moderate computational complexity (≈5.8 GFLOPs and ≈21.8 M parameters), lower than VGG16 (15.3 GFLOPs, 138 M parameters) and comparable to ResNet50 and standard InceptionV3. This pattern aligns with observations made by Ferentinos (2018) and Ghosal et al. (2021), who indicate that structural optimization combined with an adequate preprocessing stage can improve accuracy without unnecessarily increasing computational load [6], [16].

Furthermore, the proposed model achieved the lowest inference time (0.030 s/image) among all architectures, remaining far below VGG16 (0.201 s/image) and standard InceptionV3 (0.272 s/image). This balance between accuracy and latency meets the operational requirements of embedded autonomous diagnostic systems in the field, where response speed is critical for real-time decision-making.

In this work, the CNN_{propia} was compared against four reference convolutional architectures (ResNet50, VGG16, MobileNetV2, and InceptionV3), covering both deep models and a lightweight architecture designed for edge devices (MobileNetV2). However, recent variants such as Efficient Net-Lite and YOLO-nano detectors were not included, despite their potential to offer even more favourable trade-offs between accuracy and latency, particularly on hardware such as Jetson Nano or other higher-capacity embedded modules.

Similarly, the comparison focused on deep networks; traditional classifiers (e.g., SVM or Random Forest over handcrafted descriptors) and an explicit human visual inspection baseline were not incorporated. The absence of these references limits direct comparison with non-deep methods and conventional agronomic practice. Extending the analysis to ultralight architectures (Efficient Net-Lite, YOLO-nano), classical classifiers, and expert phytopathologist evaluations has been identified as necessary future work to complete the benchmarking of the system.

4.2. Performance Under Real Conditions of Illumination and Background

Although the tests were conducted primarily under controlled conditions, the system demonstrated resilience to mild lighting variations and heterogeneous backgrounds, reflected in the stability of performance during cross-validation (standard deviation of ±2.12% for the proposed model). This behaviour is consistent with findings by Wu et al. (2023) [9], who showed that diversification of the training set and chromatic normalization improve model robustness against environmental changes.

The stability of the results suggests that the proposed pipeline is capable of generalizing adequately in scenarios where visual conditions are not uniform, thereby reducing the gap traditionally observed between laboratory results and the demands of agricultural field conditions, a limitation widely reported in the literature [7], [9].

4.3. Comparison with Recent Studies

The obtained values place the system within the upper performance range reported in recent diagnostic studies published in *Computers and Electronics in Agriculture* and *Plant Phenomics*, where hybrid models and heavier architectures achieve similar accuracy levels but require substantially more computational resources [6], [10].

A key differentiating aspect is the ability to perform local inference on a Raspberry Pi 4 without reliance on cloud services. This aligns with the agricultural edge computing approach described by Bashir et al. (2022) [42], in which autonomy, latency reduction, and elimination of communication costs are critical factors for the adoption of intelligent systems in rural areas with limited connectivity.

4.4. Limitations and Opportunities for Improvement

Despite the encouraging results, the system presents limitations that open new avenues for future work:

• The dataset is based exclusively on RGB images, which limits the detection of subclinical symptoms. Incorporating hyperspectral or multitemporal imagery could improve diagnostic sensitivity [13], [28].

• The system has not yet undergone exhaustive testing under partial occlusions, strong shadows, or adverse weather conditions.

• Although the network was optimized for embedded inference, extending the system to multi-node or cooperative configurations across agricultural robots will require new communication and synchronization mechanisms. Future integration with 3D robotic vision, thermal sensors, or LiDAR could enhance the system's ability to perform multilayer diagnostics and autonomous phytosanitary control, following

the automation trends proposed by Oberti et al. (2016) [32] and Bac et al. (2014) [36].
• Although the dataset includes leaves of different orientations and sizes and data augmentation was applied (rotations, flips, brightness/contrast adjustments), no dedicated experimental protocol was designed to quantify robustness under severe occlusions, leaf overlap, or extreme scale variation. Evaluating these scenarios remains pending and is necessary to ensure stable behaviour in highly dynamic field conditions.
• Tests were conducted on images captured under field conditions, but a long-term evaluation campaign with the system deployed on the mobile robotic platform in a commercial crop has not yet been carried out. Continuous-operation validation is considered an essential phase for assessing the integrated performance of perception, navigation, and actuation in real agricultural environments.

4.5. Relevance of the Proposed Approach

The system presented constitutes a significant advancement toward sustainable digital agriculture, integrating computer vision, embedded artificial intelligence, and the potential for robotic deployment into a single accessible platform. The results demonstrate the feasibility of combining low-cost hardware—such as Raspberry Pi and inexpensive cameras—with optimized high-performance models, expanding access to intelligent diagnostic technologies for small and medium-sized producers.
The proposed approach contributes to the democratization of agricultural AI and lays the foundation for developing autonomous solutions capable of sustained operation in rural environments with limited resources. This positions the system as a promising tool for continuous health monitoring and targeted treatment application, promoting more efficient and sustainable phytosanitary practices.

V. CONCLUSIONS

The development of this system allowed us to verify that it is indeed possible to deploy deep learning models on embedded platforms without compromising diagnostic quality. The proposed model—with HSV-based segmentation and a fine-tuning process—achieved the best performance in the entire study. It reached an Accuracy of 95.67% ± 2.12% and an inference time of only 0.030 s per image on the Raspberry Pi 4, confirming that its performance is not only competitive but fully suitable for real field use, where response speed and computational autonomy are critical.
In practical terms, an accuracy of 95.67% ± 2.12% implies that out of every 100 leaves evaluated, approximately 96 are correctly classified by the system. Compared to traditional visual inspection—which typically relies on limited subsampling and strongly depends on the technician's expertise—the proposed approach provides more consistent criteria and extends diagnosis to a much larger proportion of the crop. This leads to earlier and more systematic detection of infection hotspots, especially in contexts where access to phytopathology specialists is limited or expensive.
Beyond the numerical results, the methodological process we followed provides a clear path for future developments. The combination of well-designed chromatic preprocessing, careful parameter selection, and progressive fine-tuning enabled the system to maintain stability across folds and a healthy balance between complexity and accuracy. This opens the possibility for similar technologies to be adapted to other crops or diseases, even under resource-limited settings.
The study also exposed several limitations. The exclusive use of RGB imagery imposes restrictions when detecting very early symptoms or physiological changes that are not visually apparent. Additionally, most tests were conducted under controlled or semi-stable conditions; performance under stronger illumination variations, occlusions, rain, or rapidly changing image quality remains to be evaluated.

Exploring distributed or multi-node configurations—where several devices cooperate—also remains pending and would be valuable for large-scale crop monitoring.
Throughout development, we faced significant challenges: from the natural variability of plant material to the need to adapt the architecture to operate smoothly on hardware as limited as the Raspberry Pi. Adapting the pipeline, managing data, and optimizing processing were key learning outcomes that strengthened the system.

Overall, the results show that it is entirely feasible to build fast, accurate, and accessible phytosanitary diagnostic solutions using optimized models and low-cost hardware. This project establishes a solid foundation for advancing toward more autonomous agricultural platforms, integrating computer vision, embedded AI, and actuation systems that enable localized, real-time treatment application. Looking ahead, expanding the dataset, incorporating modalities such as hyperspectral imaging, and integrating the system with robots or UAVs represent promising research directions that could elevate this approach to an industrial level.

VI. LIMITATIONS

This study was conducted using a curated image dataset compiled from publicly accessible sources; therefore, acquisition conditions such as illumination, background, camera type, and severity labelling criteria were not fully standardized. While the evaluated models showed promising results, additional validation with field-acquired images across multiple locations, crop growth stages, and environmental conditions is necessary to determine the generalizability and practical reliability of the proposed system in real-world applications.

VII. FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

VIII. DATA AVAILABILITY STATEMENT

The datasets analysed for training the convolutional neural networks were obtained from publicly available third-party repositories: Mendeley Data, *Data for: Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural*

Network, DOI: **10.17632/tywbtsjrjv.1**, available at **https://data.mendeley.com/datasets/tywbtsjrjv/1**, and the Kaggle Plant Village dataset, available at **https://www.kaggle.com/datasets/emmarex/plantdisease**.

The field-test images and experimental records generated during the current study during the real-world evaluation of the embedded-vision system are available from the corresponding author on reasonable request.

IX. REFERENCES

[1] P. W. Crous, A. Y. Rossman, M. C. Aime, W. C. Allen, T. Burgess, J. Z. Groenewald y L. A. Castlebury, "Names of Phytopathogenic Fungi: A Practical Guide," *Phytopathology*, vol. 111, no. 9, pp. 1500–1508, 2021. Available in: https://doi.org/10.1094/PHYTO-11-20-0512-PER. [Accessed: sep. 15, 2025].

[2] J. H. J. Leveau, "Re-envisioning the Plant Disease Triangle," *Annual Review of Phytopathology*, vol. 62, 2024. Available in: https://doi.org/10.1146/annurev-phyto-121423-042021. [Accessed: oct. 11, 2025].

[3] R. N. Strange y P. R. Scott, "Plant disease: a threat to global food security," *Annual Review of Phytopathology*, vol. 43, pp. 83–116, 2005. Available in: https://doi.org/10.1146/annurev.phyto.43.113004.133839. [Accessed: sep. 29, 2025].

[4] W. E. Fry, "*Phytophthora infestans*: New Tools (And Old Ones) Lead to New Understanding and Precision Management," *Annual Review of Phytopathology*, vol. 54, pp. 529–547, 2016. Available in: https://doi.org/10.1146/annurev-phyto-080615-095951. [Accessed: ago. 28, 2025].

[5] S. P. Mohanty, D. P. Hughes y M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016. Available in: https://doi.org/10.3389/fpls.2016.01419. [Accessed: sep. 15, 2025].

[6] K. P. Ferentinos, "Deep Learning Models for Plant Disease Detection and Diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018. Available in: https://doi.org/10.1016/j.compag.2018.01.009. [Accessed: oct. 11, 2025].

[7] A. Kamilaris y F. X. Prenafeta-Boldú, "Deep Learning in Agriculture: A Survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018. Available in: https://doi.org/10.1016/j.compag.2018.02.016. [Accessed: sep. 29, 2025].

[8] J. G. A. Barbedo, "A Review on the Use of Machine Learning for Image-Based Plant Disease Detection," *Agricultural Systems*, vol. 165, pp. 102–115, 2018. Available in: https://doi.org/10.1016/j.agsy.2018.05.012. [Accessed: ago. 28, 2025].

[9] X. Wu, X. Fan, P. Luo, S. D. Choudhury, T. Tjahjadi y C. Hu, "From Laboratory to Field: Unsupervised Domain Adaptation for Plant Disease Recognition in the Wild," *Plant Phenomics*, 2023. Available in: https://doi.org/10.34133/plantphenomics.0038. [Accessed: sep. 15, 2025].

[10] C. Qi, X. Zhao, X. Sun, C. Wang y Q. Zhang, "In-field Classification of the Asymptomatic Biotrophic Phase of Potato Late Blight of Leaves in Greenhouses from Hyperspectral Measurements," *Computers and Electronics in Agriculture*, vol. 206, p. 107585, 2023. Available in: https://doi.org/10.1016/j.compag.2022.107585. [Accessed: oct. 11, 2025].

[11] X. Zhang, Z. Zhang, X. Sun, Y. Yu y L. Chen, "Hyperspectral Imaging Analysis for Early Detection of Tomato Bacterial Leaf Spot Disease," *Scientific Reports*, 2024. Available in: https://doi.org/10.1038/s41598-024-78650-6. [Accessed: sep. 29, 2025].

[12] C. Xie, J. Yang, L. He, Y. Feng y L. Liu, "Detection of Early Blight and Late Blight Diseases on Tomato Leaves Using Hyperspectral Imaging," *Scientific Reports*, vol. 5, p. 16564, 2015. Available in: https://doi.org/10.1038/srep16564. [Accessed: ago. 28, 2025].

[13] A. A. Terentev, A. N. Dolzhenko y I. V. Dolzhenko, "Current State of Hyperspectral Remote Sensing for Early Plant Disease Detection," *Remote Sensing Applications: Society and Environment*, vol. 25, p. 100676, 2022. Available in: https://doi.org/10.1016/j.rsase.2021.100676. [Accessed: sep. 15, 2025].

[14] H. De Chant, M. A. Merzlyak, A. A. Melikhova, D. L. Van der Heijden, J. P. Kramer y D. Di Tommaso, "Automated Identification of Plant Disease from Mobile Imagery with Region Average Pooling," *Plant Methods*, vol. 17, p. 2, 2021. Available in: https://doi.org/10.1186/s13007-020-00705-2. [Accessed: oct. 11, 2025].

[15] J. Ramcharan, P. McCloskey, K. Baranowski, S. Mbilinyi, J. Mrisho, L. Ndalahwa, J. Legg y D. P. Hughes, "A Mobile-Based Deep Learning Model for Cassava Disease Diagnosis," *Frontiers in Plant Science*, vol. 10, p. 272, 2019. Available in: https://doi.org/10.3389/fpls.2019.00272. [Accessed: sep. 29, 2025].

[16] M. Ghosal, D. Blystone, A. K. Singh, B. Ganapathysubramanian, A. Singh y S. Sarkar, "An Explainable Deep Learning Framework for Plant Disease Classification," *IEEE Access*, vol. 9, pp. 154751–154761, 2021. Available in: https://doi.org/10.1109/ACCESS.2021.3127432. [Accessed: ago. 28, 2025].

[17] S. Too, L. Yujian, S. Njuki y L. Yingchun, "A Comparative Study of Fine-Tuning Deep Learning Models for Plant Disease Identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019. Available in: https://doi.org/10.1016/j.compag.2018.03.032. [Accessed: sep. 15, 2025].

[18] K. He, X. Zhang, S. Ren y J. Sun, "Deep Residual Learning for Image Recognition," *Proc. CVPR*, pp. 770–778, 2016. Available in: https://doi.org/10.1109/CVPR.2016.90. [Accessed: oct. 11, 2025].

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens y Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proc. CVPR*, pp. 2818–2826, 2016. Available in: https://doi.org/10.1109/CVPR.2016.308. [Accessed: sep. 29, 2025].

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov y L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proc. CVPR*, pp. 4510–4520, 2018. Available in:

https://doi.org/10.1109/CVPR.2018.00474. [Accessed: ago. 28, 2025].
[21] K. Simonyan y A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Proc. ICLR*, 2015. Available in: https://arxiv.org/abs/1409.1556. [Accessed: sep. 15, 2025].
[22] A. Krizhevsky, I. Sutskever y G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. Available in: https://doi.org/10.1145/3065386. [Accessed: oct. 11, 2025].

[23] N. Dvornik, C. Schmid y J. Mairal, "On the Importance of Visual Context for Data Augmentation in Scene Understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 2014–2028, 2021. Available in: https://doi.org/10.1109/TPAMI.2019.2958168. [Accessed: sep. 29, 2025].
[24] N. D. Lane, P. Warden, M. Abadi, A. Ramirez, C. Hall, S. Gowda y Y. Zhou, "DeepX: Resource-Efficient Deep Learning on Mobile Devices," *Proc. IPSN*, pp. 1–12, 2016. Available in: https://doi.org/10.1109/IPSN.2016.7460664. [Accessed: ago. 28, 2025].
[25] Z. Wang, M. Li, L. Luo y H. Wang, "Deploying CNNs on Raspberry Pi: A Benchmarking Study for Real-Time Image Classification," *IEEE Access*, vol. 8, pp. 193309–193322, 2020. Available in: https://doi.org/10.1109/ACCESS.2020.3032909. [Accessed: sep. 15, 2025].
[26] A. A. Mahlein, "Plant Disease Detection by Imaging Sensors—Parallels and Specific Demands for Precision Agriculture and Plant Phenotyping," *Plant Disease*, vol. 100, no. 2, pp. 241–251, 2016. Available in: https://doi.org/10.1094/PDIS-03-15-0340-FE. [Accessed: oct. 11, 2025].
[27] S. A. Mutka y R. S. Bart, "Image-Based Plant Phenotyping for Plant Breeding," *Trends in Plant Science*, vol. 20, no. 12, pp. 822–833, 2015. Available in: https://doi.org/10.1016/j.tplants.2015.06.003. [Accessed: sep. 29, 2025].
[28] J. B. Kuska, A. Wahabzada, A. A. Leucker, C. Dehne, H.-W. Dehne, E.-C. Mahlein y U. Steiner, "Hyperspectral Phenotyping on the Microscopic Scale: Towards Automated Characterization of Plant-Pathogen Interactions," *Journal of Plant Physiology*, vol. 176, pp. 147–156, 2015. Available in: https://doi.org/10.1186/s13007-015-0073-7. [Accessed: ago. 28, 2025].
[29] A. A. Mahlein, U. Steiner, U. Dehne y E.-C. Oerke, "A Comparison of Hyperspectral Imaging and Multispectral Fluorescence Imaging for Plant Disease Detection," *Plant Methods*, vol. 8, p. 3, 2012. Available in: https://doi.org/10.1186/1746-4811-8-3. [Accessed: sep. 15, 2025].
[30] P. Skelsey, W. van der Werf, J. A. N. van den Bosch, S. J. F. van der Werf y J. A. J. Metz, "Invasion of *Phytophthora infestans* at the Landscape Level: How Do Spatial Scale and Weather Modulate Dispersal and Disease Spread?," *Phytopathology*, vol. 99, no. 8, pp. 847–857, 2009. Available in: https://doi.org/10.1094/PHYTO-99-8-0847. [Accessed: oct. 11, 2025].
[31] A. J. Haverkort, P. C. Struik, R. G. F. Visser y E. Jacobsen, "Applied Biotechnology to Combat Late Blight in Potato Caused by *Phytophthora infestans*," *Potato Research*, vol. 59, pp. 35–66, 2016. Available in: https://doi.org/10.1007/s11540-015-9301-7. [Accessed: sep. 29, 2025].
[32] R. Oberti, M. Marchi, P. Tirelli, A. Calcante, M. Iriti, M. Hočevar, J. Baur, C. Schütz y H. Ulbrich, "Selective Spraying of Grapevines for Disease Control Using a Modular Agricultural Robot," *Biosystems Engineering*, vol. 146, pp. 203–215, 2016. Available in: https://doi.org/10.1016/j.biosystemseng.2015.12.017. [Accessed: ago. 28, 2025].
[33] Y. Chen, H. Zhu y E. Ozkan, "Development of a Variable-Rate Sprayer with Laser Scanning Sensor to Synchronize Spray Outputs to Tree Structures," *Transactions of the ASABE*, vol. 55, no. 3, pp. 773–781, 2012. Available in: https://doi.org/10.13031/2013.41509. [Accessed: sep. 15, 2025].
[34] H. Liu y H. Zhu, "Evaluation of a Laser Scanning Sensor in Detection of Complex-Shaped Targets for Variable-Rate Sprayer Development," *Transactions of the ASABE*, vol. 59, no. 5, pp. 1181–1192, 2016. Available in: https://doi.org/10.13031/trans.59.11760. [Accessed: oct. 11, 2025].
[35] J. Cai, X. Wang, Y. Gao, S. Yang y C. Zhao, "Design and Performance Evaluation of a Variable-Rate Orchard Sprayer Based on a Laser-Scanning Sensor," *International Journal of Agricultural and Biological Engineering*, vol. 12, no. 6, pp. 51–57, 2019. Available in: https://doi.org/10.25165/j.ijabe.20191206.4174. [Accessed: sep. 29, 2025].
[36] M. Bac, E. J. van Henten, J. Hemming y Y. Edan, "Harvesting Robots for High-Value Crops: State-of-the-Art Review and Challenges Ahead," *Journal of Field Robotics*, vol. 31, no. 6, pp. 888–911, 2014. Available in: https://doi.org/10.1002/rob.21525. [Accessed: ago. 28, 2025].
[37] S. Hellström, T. Ringdahl y U. Wiklund, "Autonomous Navigation in Agricultural Fields Using Vision-Based Row Detection," *Computers and Electronics in Agriculture*, vol. 175, p. 105580, 2020. Available in: https://doi.org/10.1016/j.compag.2020.105580. [Accessed: sep. 15, 2025].
[38] N. H. M. Noor, S. Ahmad, R. Mohd-Mokhtar, H. Hassan y A. A. Shafie, "Vision-Based Navigation in Orchards: A Review," *Computers and Electronics in Agriculture*, vol. 198, p. 107024, 2022. Available in: https://doi.org/10.1016/j.compag.2022.107024. [Accessed: oct. 11, 2025].
[39] M. H. Saleem, J. Potgieter y K. M. Arif, "Plant Disease Detection Using Deep Learning: A Comprehensive Survey," *Sustainable Computing: Informatics and Systems*, vol. 28, p. 100407, 2020. Available in: https://doi.org/10.1016/j.suscom.2019.100407. [Accessed: sep. 29, 2025].
[40] M. Sokolova y G. Lapalme, "A Systematic Analysis of Performance Measures for Classification Tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009. Available in: https://doi.org/10.1016/j.ipm.2009.03.002. [Accessed: ago. 28, 2025].
[41] T. Saito y M. Rehmsmeier, "The Precision-Recall Plot Is More Informative Than the ROC Plot When Evaluating Binary

Classifiers on Imbalanced Datasets," *PLOS ONE*, vol. 10, no. 3, e0118432, 2015. Available in: https://doi.org/10.1371/journal.pone.0118432. [Accessed: sep. 15, 2025].

[42] T. A. M. Bashir, H. Qureshi, M. S. Sarfraz, M. L. Baazaoui y N. Conci, "Edge Computing for AIoT: Vision, Enabling Technologies, and Open Challenges," *IEEE Access*, vol. 10, pp. 84564–84598, 2022. Available in: https://doi.org/10.1109/ACCESS.2022.3198669. [Accessed: oct. 11, 2025].