Detection of Maize Foliar Diseases Using AI Optimized for Deployment on Edge Devices

Oscar J. González Zárate[1,2], Luis E. Macea Zabaleta[1], Andrés F. Flórez Olivera[1], Nicolás Castillo Ojeda[1], Taciano Ares Rodolfo[2,] Carlos J. Gonzalez Aguilera[2]

[1] Universidad de los Andes, Bogotá – Colombia

[2] Center for Embedded Devices and Research in Digital Agriculture (CEDRA), São Leopoldo - RS, Brasil

o.gonzalezz@uniandes.edu.co, l.macea@uniandes.edu.co, a.florezo@uniandes.edu.co, n.castilloo@uniandes.edu.co, taciano.rodolfo@senairs.org.br, carlos.gonzalez@senairs.org.br,

A paper from the Proceedings of the

17th International Conference on Precision Agriculture and 11th Brazilian Congress on Precision and Digital Agriculture

13-15 July 2026

Porto Alegre, Brazil

Abstract.

Maize plays a central role in food security across tropical and temperate regions. Its productivity is significantly affected by several foliar diseases, among which—common rust, gray leaf spot, and blight—are some of the most prevalent and damaging. These pathologies can cause substantial yield losses if not detected and treated in a timely manner, making early diagnosis a fundamental factor to ensure healthy and sustainable crop development. However, traditional diagnostic methods based on manual visual inspection are often slow, subjective, and dependent on expert knowledge, which limits their applicability in rural areas with restricted connectivity and technological resources.

At the same time, the evolution of the agro-industrial sector has promoted the adoption of strategies aimed at sustainability, focusing on optimizing water usage, reducing pesticide application, and mitigating environmental impact. Within this framework, precision agriculture has emerged as a structured approach that integrates digital technologies, sensors, and data-driven analysis to support efficient and localized agronomic decision making.

This work presents the development of an automated maize foliar disease diagnostic system based on computer vision, specifically designed for deployment on edge computing devices. The proposed system is intended to operate without cloud dependency and under constrained computational resources, allowing local and offline inference in environments with limited connectivity. The solution consists of a multiclass classifier optimized for embedded execution.

The methodology includes a comparative evaluation of four deep learning architectures—MobileNetV3-Large, EfficientNet-LiteB0, MobileViT, and Plant Mobile Vision Transformer (PMVT)—using transfer learning with a progressive fine-tuning strategy. Model training was implemented in Python using TensorFlow, while the complete lifecycle of each model was tracked through MLflow to ensure experimental traceability. Edge feasibility was

assessed by measuring global accuracy, class- wise recall, and inference latency.

The experimental results indicate that MobileNetV3-Large provides the best trade-off between predictive performance and computational efficiency. A classification head was added to the ImageNet-pretrained backbone, and the model was trained using a dataset of 6,952 images, achieving a test accuracy approaching to ≈ 0.92 and a class-wise recall greater than >= 0.83. Consequently, this model was selected for further optimization through structural pruning and dynamic range quantization techniques. The optimized model preserved a comparable level of accuracy (≈ 0.92) while achieving a substantial reduction in model size and inference latency, thus meeting embedded hardware constraints. These results enable fast, objective, and autonomous in-field diagnostics, supporting the adoption of precision agriculture practices in regions with limited connectivity.

Keywords.
Precision agriculture; Computer vision; Deep learning; Edge computing; Transfer learning

Introduction

Global maize production faces recurring losses from foliar diseases. Its productivity is affected by diverse foliar diseases capable of generating significant yield losses (Mueller et al. 2020). Traditional detection, based on visual inspection by experts, is subjective, slow, and difficult to scale in extensive crops (Mohanty et al. 2016). Subjective diagnosis delays interventions and increases fungicide misuse, increasing costs and reducing productive efficiency **(**Mohanty et al. 2016**;** Mueller et al. 2020**)**.

These limitations are compounded by the fact that many current digital solutions require stable connectivity or high computational resources, operational conditions that are often unaffordable in rural areas, especially in South America (Abiri et al. 2023; Basso and Antle 2020). Edge computing removes the need for cloud connectivity by executing near-real-time inference locally on embedded hardware, reducing infrastructure costs, and facilitate distributed deployments in environments with restricted connectivity (O'Grady et al. 2019; Singh and Gill 2023). This approach is particularly relevant for precision agriculture applications where the goal is to support producers with efficient, portable systems that can operate outside the cloud.

Given these challenges, it is necessary to advance toward computer-vision models that maintain an adequate level of diagnostic accuracy while also being able to run on edge devices with limited resources. Likewise, there remains a need to systematize data curation processes, evaluate efficient state-of-the-art architectures, and apply compression methods—such as pruning and quantization—to reduce the computational footprint without compromising performance (Singh and Gill 2023).

In response, this study introduces a deep-learning classifier for maize foliar diseases optimized for edge execution. The study integrates data from public repositories, evaluates modern deep learning architectures under transfer learning, and explores optimization techniques to enable inference without cloud dependence. The results allow analysis of the potential of an edge approach to support phytopathological diagnosis in agricultural regions.

Contributions

This work provides:

i. A reproducible data preparation protocol with perceptual hashing and class balancing.
ii. a benchmark of edge-oriented models.
iii. a reproducible architecture and training process via transfer learning; and
iv. a post-training optimization study comparing pruning, quantization, and their combination.

These interconnected contributions establish a validated baseline for deploying offline phytopathological monitoring systems in infrastructure-limited environments.

Methodology

Processing and system development followed an edge-computing-oriented approach, prioritizing lightweight models with efficient architectures and low latency, consistent with the current literature in digital agriculture (Abiri et al. 2023; Basso and Antle 2020). The solution backbone was restricted to architectures specifically designed for limited-resource environments: *MobileNetV3-Large*, *EfficientNet-LiteB0*, *MobileViT*, and *PMVT*. These networks feature a reduced number of parameters and sizes compatible with on-device deployment, facilitating transfer learning and later optimization via pruning and quantization (Craze et al. 2022; Li et al. 2023; Qian et al. 2022).

Transfer learning was used to accelerate convergence and improve generalization using agricultural-domain data (Chollet 2021). To ensure device viability, the methodology includes a two-stage post-training optimization process: first pruning, then quantization. These techniques were evaluated in a controlled manner to measure their impact on accuracy, class-wise recall, and latency. Finally, the resulting model was converted to TensorFlow Lite format, producing a lightweight executable (.tflite) suitable for edge devices.

A. Data collection

The dataset was built using RGB images of maize leaves from two open repositories, totaling 7,835 images. This total includes 4,188 images extracted from the *Corn Leaf Disease Dataset*. This dataset is based on previously published as *PlantVillage* and *PlantDoc* (Pandian and Geetharamani 2019; Singh et al. 2020), consolidated through a public *Kaggle repository* (Ghose n.d.) and 3,647 images from the *Corn Diseases dataset* (Roboflow Universe n.d.).

B. Exploratory data analysis

Following the nomenclature of reference works (Mohanty et al. 2016; Qian et al. 2022), labels were taxonomically unified into four classes: Healthy, Common_Rust, Gray_Leaf_Spot, and Blight. This standardization avoids ambiguities between sources and facilitates comparative evaluation. Perceptual hashing was also applied to identify and remove duplicates, preventing information leakage between partitions.

During exploration, critical limitations were identified in the original sources. The Roboflow dataset included images previously augmented with rotations and reflections without explicit documentation and completely lacked the Healthy class, preventing access to the full planned volume. Because of this pre-augmentation, this material was not considered suitable for testing and was carefully restricted to training only. On the other hand, the raw Kaggle dataset showed significant imbalance, where the minority class Gray_Leaf_Spot had only 574 images, representing a difference of over 40% compared with the other classes.

C. Preprocessing

To ensure reproducibility and obtain an unbiased performance estimate, a stratified split was performed into training (train), validation (val), and test folders. Given the disparity between sources, specifically Roboflow images with prior augmentation and Kaggle images without it, a differentiated composition strategy was designed. Validation and test subsets used only the Kaggle dataset as a base, since it contains original images without artificial alterations, ensuring evaluation on real data.

Conversely, data augmentation was applied to the training subset to correct imbalance and improve generalization. Importantly, chromatic alterations were excluded, limiting transformations to spatial changes (rotations, translations, and reflections). This decision was based on the fact that chromatic uniformity is a key diagnostic factor in foliar pathologies; modifying color, hue, or contrast could introduce artifacts that degrade the model's ability to identify chlorosis or specific lesions, a risk documented in field studies (Craze et al. 2022).

As a result, the final dataset comprised 9,928 images with a fully balanced distribution to avoid bias toward majority classes, as detailed in (Table 1). This final structure allocated 1,488 images for testing and 1,488 for validation (372 per class in each case), with 6,952 images for training (1,738 per class).

Table 1. Image distribution by class and folders after preprocessing

Class	Train	Val	Test
Common Rust	1738	372	372
Gray Leaf Spot	1738	372	372
Blight	1738	372	372
Healthy	1738	372	372

D. Feature pipeline

The input pipeline was built using TensorFlow's *image_dataset_from_directory*, generating a labeled, batched *tf.data.Dataset*. All images were resized to 224 × 224 pixels to ensure compatibility with mobile backbones and control memory usage. Normalization was then applied by rescaling intensity values to the [0, 1] range to reduce numerical variability, stabilize gradients, and favor convergence. To optimize computational performance, *prefetch(tf.data.AUTOTUNE)* was implemented, overlapping GPU processing with CPU data preparation. The class name array (*class_names*) was preserved before any optimization to ensure correct traceability in confusion matrices.

1) Model selection and training

Four state-of-the-art architectures oriented to mobile scenarios were evaluated: *MobileNetV3-Large*, *EfficientNet-LiteB0*, *MobileViT*, and *PMVT*. Selection was based on their demonstrated efficiency in balancing accuracy, latency, and memory (Craze et al. 2022; Li et al. 2023; Qian et al. 2022). Transfer learning with *ImageNet* weights was adopted to maximize data efficiency. The standardized experimental protocol was deployed on Google Colab platform with a NVIDIA T4/L4 GPU hardware acceleration, using the Adam optimizer (lr = 1e−4), batch size of 64, and an initial horizon of 20 epochs. EarlyStopping, ReduceLROnPlateau, and ModelCheckpoint callbacks were configured.

In the selection phase, whose objective was to identify the most robust behavior rather than hyperparameter optimization, each architecture was subjected to multiple configurations. (Fig.1) presents the comparative statistical analysis derived from these tests.

Results showed that *MobileNetV3-Large* offered the best overall performance. It achieved an average accuracy of 0.915 with the lowest standard deviation (Fig.1b-c), outperforming *EfficientNet* (0.755), *MobileViT* (0.718), and *PMVT* (0.641), and presented the best test loss/accuracy relationship (Fig.1d). For the critical class Gray_Leaf_Spot, *MobileNetV3-Large* achieved a minimum recall of 0.853, exceeding the 0.85 target. It also presented a significantly smaller memory footprint (13 MB versus 33 MB for the other models) and greater stability across runs. Therefore, *MobileNetV3-Large* was selected as the base architecture.

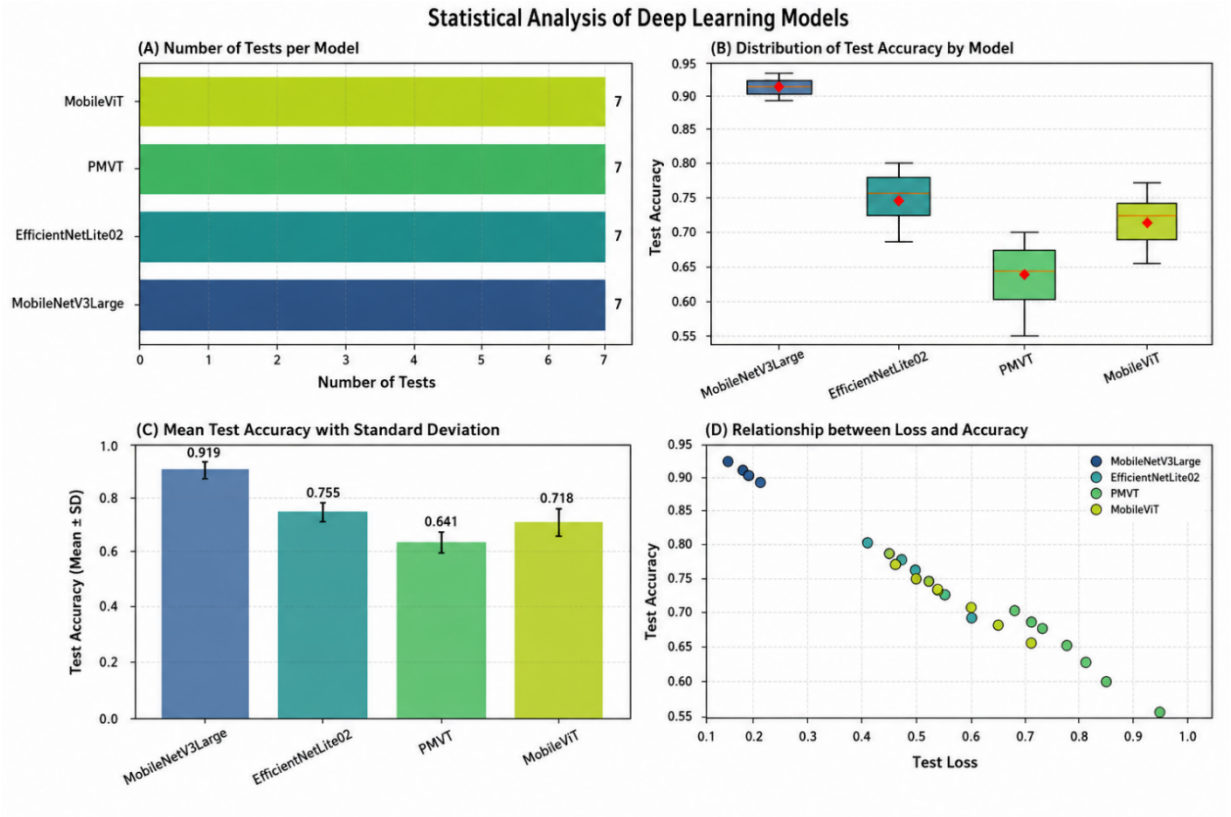

Fig 1. Comparative statistical analysis of the architectures. (a) Evaluations per model. (b) Accuracy distribution. (c) Mean accuracy and standard deviation. (d) Loss/accuracy relationship.

E. Training MobileNetV3-Large

A two-phase training pipeline was developed. The first "warm-up" phase trained with the backbone frozen for 5 epochs to stabilize the new top layers. The second fine-tuning phase progressively unfroze from layer 100 onward, reduced the learning rate by ×10, and extended training for 15 additional epochs.

The final architecture uses *MobileNetV3-Large* (pretrained on ImageNet, *include_top=False*, *pooling='avg'*) followed by a domain-specific classification head. This head includes two dense layers (512 and 256 neurons) with Batch Normalization and Dropout regularization (0.5 and 0.3) and ends with a 4-neuron Softmax layer. This configuration adds 625,924 trainable parameters (Table 2).

Table 2. Classification head architecture details and parameters

Layer (type)	Output Shape	Params
MobileNetV3Large (Functional)	(None, 960)	2,996,352
dense (Dense)	(None, 512)	492,032
batch normalization	(None, 512)	2,048
dropout	(None, 512)	0
dense_1 (Dense)	(None, 256)	131,328
batch normalization_1	(None, 256)	1,024
dropout_1	(None, 256)	0
dense_2 (Dense)	(None, 4)	1,028

Training and validation curves showed adequate convergence. The final model reached an average accuracy of 93.08% and class-wise recall around 93.01% (Table 3). The resulting *.keras* file size was 40.78 MB, validating preliminary suitability for edge devices (Singh and Gill 2023).

Table 3. Detailed performance metrics in training and validation.

	precision	recall	f1-score	support

Blight	0.9017	0.8387	0.8691	372
Common_Rust	0.9811	0.9785	0.9798	372
Gray_Leaf_Spot	0.8485	0.9032	0.8750	372
Healthy	0.9920	1.0000	0.9960	372
accuracy			0.9301	1488
macro avg	0.9308	0.9301	0.9300	1488
weighted avg	0.9308	0.9301	0.9300	1488

The test confusion matrix (Fig. 2) revealed excellent performance for Healthy and Common_Rust, while residual errors concentrated in morphological overlap between Blight and Gray_Leaf_Spot, a challenge documented in the literature (Craze et al. 2022).

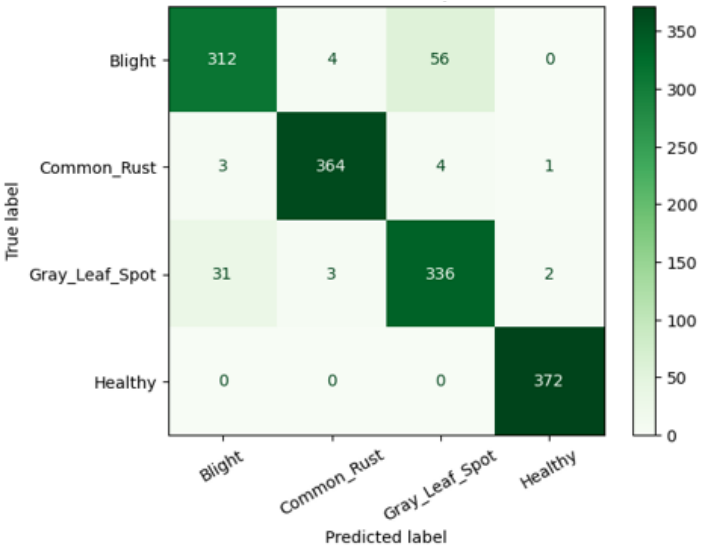

Fig 2. Confusion matrix of the test set applied to the final MobileNetV3-Large model.

F. Edge optimization: pruning and quantization

To reduce memory footprint and latency without degrading diagnostic capacity (Cherubin et al. 2022), (Bassoi et al. 2024), (Singh and Gill 2023), optimization techniques were applied. The procedure began with pruning: the base model was cloned and a target sparsity of 50% was applied to dense layers to remove redundant weights. A second version of the pruned model then underwent brief fine-tuning to recover potential accuracy loss, followed by dynamic range quantization (DRQ). DRQ reduces the precision of float32 weights and activations to 8-bit integers (int8), decreasing model size with minimal accuracy impact.

Three final quantized versions were generated and evaluated under standard metrics (accuracy, precision, recall, f1-score, size in MB, and inference time in ms):

(1) the quantized original model,
(2) the pruned and quantized model, and
(3) the pruned + fine-tuned + quantized model.

Results and Evaluation

A series of tests was implemented on models obtained in each process stage using the test set (372 images per class), which contains no augmented images and was not used during training. Learning and validation curves showed consistent behavior without signs of premature overfitting, supporting progression to edge-oriented optimization.

A. Pruning

Accuracy shifted by exactly 0.27 percentage points, from 0.9234 in the uncompressed model to 0.9207 post-prunning, as shown in (Fig. 3). Resource improvements were substantial: file size was reduced by 64.5% (from 40.78 MB to 14.49 MB), and mean latency per image decreased from 29.41 ms to 22.81 ms. Total sparsity reached 49.9%.

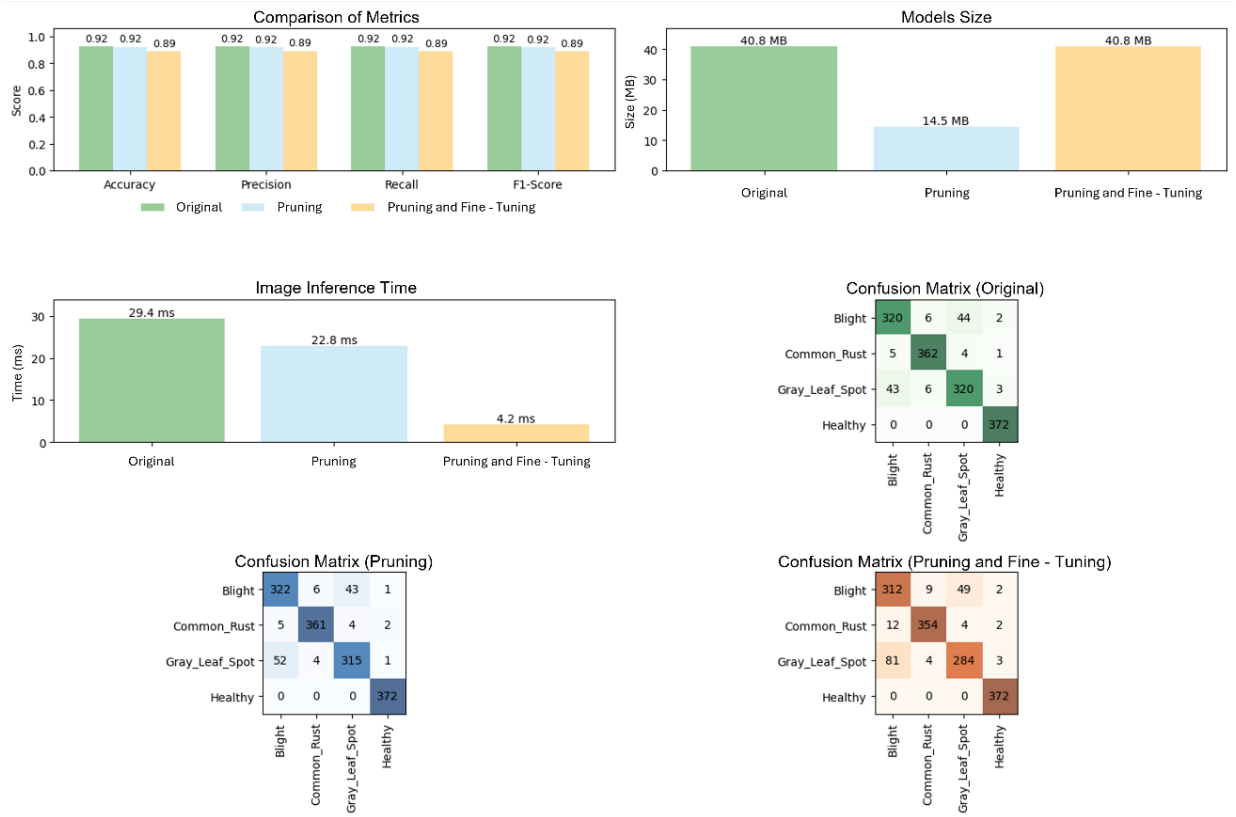

Fig 3. Comparison of results for the original, pruned, and fine-tuned pruned models.

A brief fine-tuning on the pruned model drastically reduced latency to 4.19 ms but caused notable performance degradation: accuracy dropped to 0.8884 and precision/recall/f1-score decreased by 3.3–3.5 percentage points. The file size remained at 40.78 MB because physical size reduction requires a stripping stage that was not effective after retraining.

Confusion matrices across the three models showed consistent patterns: Healthy and Common_Rust remained strong, while errors concentrated between Blight and Gray_Leaf_Spot.

Conclusion for this stage: the pruned model without fine-tuning was the most balanced candidate, improving memory footprint and latency while preserving accuracy.

B. Quantization

Dynamic range quantization was applied to three routes: the original model, the pruned model, and the pruned + fine-tuned model.

1) Metrics and size analysis

Quantization proved low-variance for preserving performance in both the original and pruned models, maintaining aggregated metrics around 0.92 as shown in (Fig. 4). The pruned + fine-tuned variant showed a generalized drop near 0.89.

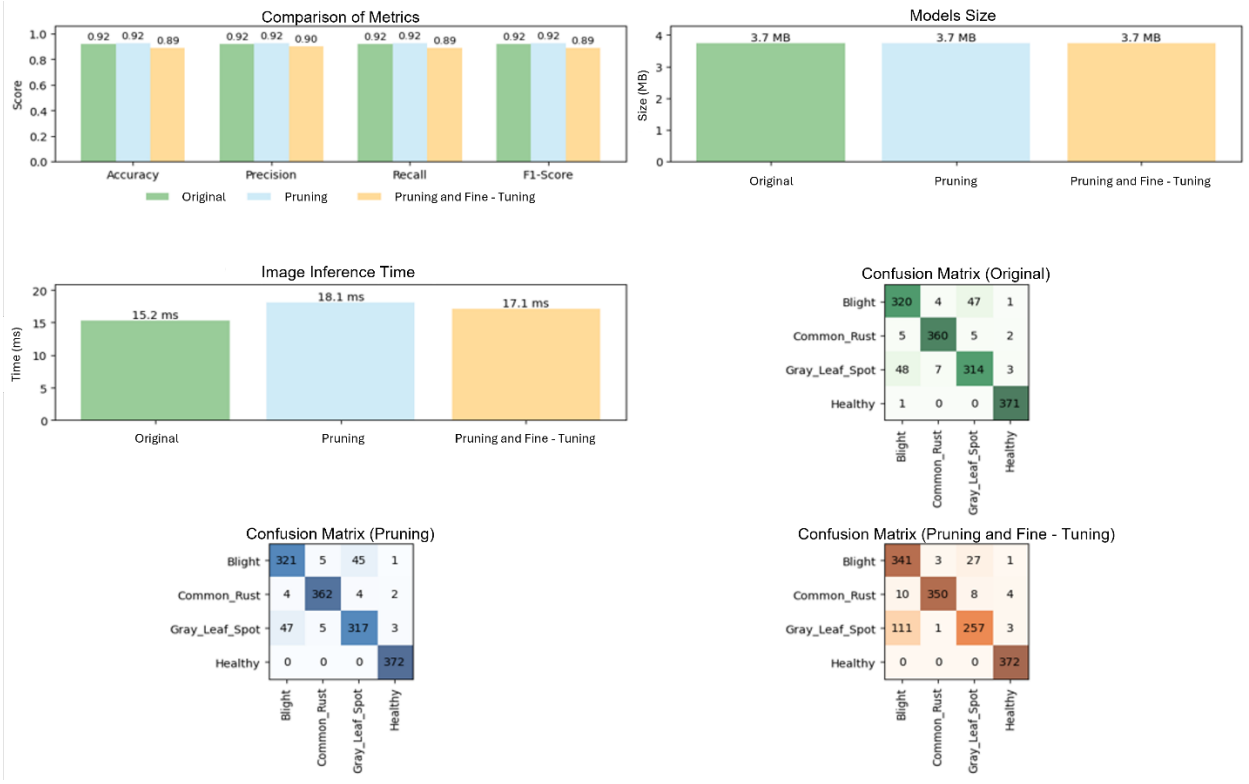

Fig 4. Comparison of quantization results across the original, pruned, and pruned-with-fine-tuning models.

Empirical metrics demonstrated structural storage-size convergence: after conversion to .tflite, all three models had approximately the same size (3.7 MB). This indicates that, under DRQ, prior pruning did not add significant additional benefit regarding final executable memory footprint.

Table 4. Comparative performance after quantization.

Model	Accuracy	Precision	Recall	F1-score
Original	0.92	0.92	0.92	0.92
Pruned	0.92	0.92	0.92	0.92
Pruned + Fine-Tuning	0.89	0.90	0.89	0.89

2) Latency analysis

Counterintuitively, the quantized baseline architecture exhibited the lowest mean latency (15.2 ms) outperforming the pruned variants. Pruned variants were slightly slower. These tests were performed with XNNPACK disabled for stability purposes.

Table 5. Average latency per image for quantized models.

Model	Latency per image (ms)
Original	15.2
Pruned	18.1
Pruned + Fine-Tuning	17.1

3) Class-wise behavior and discussion

Quantized confusion matrices confirmed stability for the original and pruned models. Healthy detection was nearly perfect and Common_Rust remained highly accurate. The principal confusions persisted between Blight and Gray_Leaf_Spot. In the pruned + fine-tuned model, although Blight detection slightly improved, Gray_Leaf_Spot separability degraded in particular.

4) Final selection

Quantization captured most size-reduction benefits (down to ~3.7 MB) without the need for prior pruning. Given that pruning did not improve latency or reduce size further after quantization, and that fine-tuning harmed accuracy, the quantized original model represents the most balanced

alternative for edge deployment.

Discussion

Our results demonstrate that dynamic range quantization alone yields a 3.7 MB model with 92% accuracy, making cloud-free inference practical on embedded hardware. However, pruning before quantization provided no additional benefit, contradicting common assumptions in edge AI literature. The experimental data validates the feasibility of localized diagnostics while revealing critical insights into dataset structural integrity and post-training optimization trade-offs.

Data unification, duplicate control via perceptual hashing, and strict separation between original and augmented images in validation/testing reduce leakage risk and provide more realistic estimates than studies based on ideal laboratory settings such as PlantVillage. The choice of *MobileNetV3-Large* is empirically justified: best balance of average accuracy, stability, and memory footprint, outperforming *EfficientNet-LiteB0*, *MobileViT*, and *PMVT*, including for minority/critical class sensitivity.

In native .keras format, post-training pruning achieved substantial size and latency reductions with minimal global metric penalty. However, after DRQ conversion to .tflite, models converged to similar sizes, indicating quantization alone captured nearly all disk-compression benefits. Latency results also favored the quantized original model.

Error patterns align with recent literature: Healthy and Common_Rust remain stable, while Blight–Gray_Leaf_Spot confusion persists due to morphological similarity and field lighting variability. Future improvements should prioritize targeted dataset expansion and cost-sensitive training strategies.

Conclusions

This work presented a complete workflow for designing, training, and optimizing a maize foliar disease diagnostic model oriented to edge devices. The framework formalizes a standardized workflow for data curation, transfer learning optimization, and embedded compression under realistic computing and memory constraints.

The data pipeline—with taxonomic unification, duplicate control, and stratified partitioning that strictly separates augmented material from validation/test—enabled performance estimates closer to real-use conditions. *MobileNetV3-Large* consolidated as the most suitable architecture, combining global accuracy approximately 90–93% with class-wise recall above defined thresholds and a moderate memory footprint. The main error pattern concentrated in the Blight–Gray_Leaf_Spot pair highlights the need for class-specific strategies but does not compromise overall system utility.

Pruning benefits were clear in native format, with 65% size reduction and latency improvements. However, after dynamic range quantization in TensorFlow Lite, models converged to 3.7 MB. The quantized original model provided the best overall compromise between accuracy, compact size, and response time, while the pruning + fine-tuning route proved detrimental.

From an agronomic perspective, enabling local inference on edge devices offers a concrete path to support phytosanitary diagnosis in regions with limited infrastructure, reducing dependency on specialists and improving response times for outbreaks.

Acknowledgements

This work was partially funded by the “IA na borda” project, by the Center for Embedded Devices and Research in Digital Agriculture (CEDRA) of SENAI-RS, with financial resources from PPI IoT/Manufacturing 4.0 / PPI HardwareBR of the MCTI, grant number 056/2023, signed with EMBRAPII.

References

Abiri, R., Rizan, N., Balasundram, S. K., Shahbazi, A. B., & Abdul-Hamid, H. (2023). Application of digital technologies for ensuring agricultural productivity. Heliyon, 9(12), e22601. https://doi.org/10.1016/j.heliyon.2023.e22601

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., & Stefanovic, D. (2019). Solving current limitations of deep learning based approaches for plant disease detection. Symmetry, 11(7), 939. https://doi.org/10.3390/sym11070939

Bassoi, L. H., Bernardi, A. C. de C., Vaz, C. M. P., Pires, J. L. F., Gebler, L., Jorge, L. A. de C., et al. (Eds.). (2024). Agricultura de precisão: Um novo olhar na era digital. São Carlos, Brazil: Editora Cubo. https://doi.org/10.4322/978-65-86819-38-0.1000001

Basso, B., & Antle, J. (2020). Digital agriculture to design sustainable agricultural systems. Nature Sustainability, 3(4), 254–256. https://doi.org/10.1038/s41893-020-0510-0

Cherubin, M. R., Damian, J. M., Tavares, T. R., Trevisan, R. G., Colaço, A. F., Eitelwein, M. T., et al. (2022). Precision agriculture in Brazil: The trajectory of 25 years of scientific research. Agriculture, 12(11), 1882. https://doi.org/10.3390/agriculture12111882

Chollet, F. (2021). Deep learning with Python (2nd ed.). Shelter Island, NY, USA: Manning Publications.

Craze, H. A., Pillay, N., Joubert, F., & Berger, D. K. (2022). Deep learning diagnostics of gray leaf spot in maize under mixed disease field conditions. Plants, 11(15), 1942. https://doi.org/10.3390/plants11151942

Databricks (n.d.). MLflow. Available at: https://mlflow.org (last accessed 6 November 2025).

Ghose, S. (n.d.). Corn or maize leaf disease dataset. Kaggle. Available at: https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset (last accessed 6 November 2025).

Google (n.d.). TensorFlow. Available at: https://www.tensorflow.org (last accessed 6 November 2025).

Li, G., Wang, Y., Zhao, Q., Yuan, P., & Chang, B. (2023). PMVT: A lightweight vision transformer for plant disease identification on mobile devices. Frontiers in Plant Science, 14. https://doi.org/10.3389/fpls.2023.1256773

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7. https://doi.org/10.3389/fpls.2016.01419

Mueller, D. S., Wise, K. A., Sisson, A. J., Allen, T. W., Bergstrom, G. C., Bissonnette, K. M., et al. (2020). Corn yield loss estimates due to diseases in the United States and Ontario, Canada, from 2016 to 2019. Plant Health Progress, 21(4), 238–247. https://doi.org/10.1094/PHP-05-20-0038-RS

O'Grady, M., Langton, D., & O'Hare, G. (2019). Edge computing: A tractable model for smart agriculture? Artificial Intelligence in Agriculture, 3, 42–51. https://doi.org/10.1016/j.aiia.2019.12.001

Pandian, A., & Geetharamani, G. (2019). Data for: Identification of plant leaf diseases using a 9-layer deep convolutional neural network. Mendeley Data, V1. https://doi.org/10.17632/tywbtsjrjv.1

Qian, X., Zhang, C., Chen, L., & Li, K. (2022). Deep learning-based identification of maize leaf diseases is improved by an attention mechanism: Self-attention. Frontiers in Plant Science, 13. https://doi.org/10.3389/fpls.2022.864486

Roboflow Universe (n.d.). Corn diseases dataset. Available at: https://universe.roboflow.com/corn-disease-7/corn-diseases-oxojk (last accessed 16 July 2025).

Singh, D., Gill, S. S. (2023). Edge AI: A survey. Internet of Things and Cyber-Physical Systems, 3, 71–92. https://doi.org/10.1016/j.iotcps.2023.02.004

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., & Batra, N. (2020). PlantDoc: A dataset for visual plant disease detection. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD (pp. 249–253).