Proceedings of CONF-SPML 2026 Symposium: The 2nd Neural Computing and Applications Workshop 2025
DOI: 10.54254/2755-2721/2026.TJ30950

# *Crop Disease Leaf Image Recognition System Based on CNN and Edge Computing*

**Jiani Hu**

*School of Electronics and Engineering, Beibu Gulf University, Qinzhou, China*
*40131823@qq.com*

***Abstract.*** Crop diseases represent a critical threat to global food security. Traditional manual diagnosis approaches are inefficient, while existing cloud-centric AI solutions are plagued by network latency, high data transmission costs, and data privacy risks. This study aims to design and implement an efficient, low-cost edge computing system for real-time crop disease leaf recognition. It investigates the development of a lightweight CNN model tailored for resource-constrained edge devices, its efficient deployment on a Raspberry Pi edge platform, and its advantages over cloud solutions.This paper employs a lightweight Convolutional Neural Network (CNN) based on the MobileNetV2 architecture. The model was trained on the PlantVillage dataset using transfer learning, optimized via post-training quantization, and deployed on a Raspberry Pi edge computing platform. Experimental results demonstrate that the proposed system attains an accuracy exceeding 98%, with a quantized model size of merely 3.2MB. The average inference latency on the Raspberry Pi is less than 500 milliseconds. This edge-centric solution effectively mitigates the inherent limitations of cloud-based paradigms, providing a feasible, practical, and privacy-preserving tool for in-field disease diagnosis, thereby contributing to the advancement of precision agriculture.

***Keywords:*** Edge Computing, Convolutional Neural Network, Model Deployment, Precision Agriculture, AI Model

## 1. Introduction

Plant diseases are a primary cause of crop yield loss, directly impacting economic stability and global food security [1]. The timely and accurate identification of these diseases is crucial for implementing effective control measures. Conventional methods, which rely on manual inspection by agricultural specialists, are frequently subjective, time-consuming, and lack scalability for large-scale agricultural operations.

In recent years, deep learning, particularly Convolutional Neural Networks (CNNs), has transformative potential in the domain of image-driven automated diagnosis. Seminal research by Mohanty et al. demonstrated that CNNs could achieve expert-level accuracy in the diagnosis of a diverse range of plant diseases [2]. Subsequent studies, such as that conducted by Li et al., have comprehensively documented the transformative potential of deep learning in smart agriculture [3]. However, the prevailing deployment paradigm for these high-performance models relies on cloud

DOI: 10.54254/2755-2721/2026.TJ30950

computing infrastructure. This cloud-dependent approach introduces critical limitations in agricultural settings, including the necessity for stable, high-bandwidth internet connectivity, inherent latency associated with data transmission, sustained operational costs, and potential privacy violations involving sensitive agricultural data [4].

Edge computing, a paradigm that processes data in proximity to its source, has emerged as a promising solution to address these bottlenecks by delivering low latency, improved data privacy, and reduced bandwidth consumption [5]. This paper presents a crop disease recognition system that leverages edge computing by deploying a lightweight CNN on a Raspberry Piedge platform. This research offers a practical reference framework for the application of edge AI in precision agriculture

## 2. Related work

Deep learning models have become the state-of-the-art method for plant disease recognition. CNNs automatically learn hierarchical features from images, eliminating the need for manual feature engineering. Studies have shown remarkable accuracy, they typically employ complex and computationally intensive models [2,6]. These models are designed for powerful servers and are unsuitable for direct deployment on devices with constrained computational resources, which is a significant gap for field applications.

Edge computing mitigates the limitations of cloud-centric models by performing data processing locally on devices at the network's edge. Satyanarayanan delineated the vision and inherent challenges of this architectural paradigm [5]. Its applicability in agriculture contexts has been widely acknowledged for facilitating real-time decision-making. Existing research in this area often focuses on system architectures and sensor configurations. However, there remains a need for studies that provide end-to-end implementation details, from model lightweighting to deployment on low-cost hardware like the Raspberry Pi, for specific tasks like disease recognition [7]. This research aims to fill that gap.

## 3. Methodology

### 3.1. System design and model development

The workflow commences with model development and training executed on a high-performance computing (HPC) platform to harness its superior computational capabilities. Subsequently, the optimized model transitions to edge deployment and executes real-time inference on a Raspberry Pi edge computing platform.The edge device, integrated with a camera module, acquires leaf images. The images are preprocessed and fed into the locally deployed TensorFlow Lite model. The inference outcomes, including the disease category and corresponding confidence score, are then output in real-time on the device.

### 3.2. Model development

This study employs the publicly accessible PlantVillage dataset [2]. To streamline the development workflow and validate the system's core functionality, a targeted dataset subset was curated, focusing exclusively on tomato leaves across three distinct states: Tomato Healthy, Tomato Early Blight, and Tomato Late Blight. This selection yielded a manageable yet information-rich dataset consisting of approximately 3,000 images. To ensure a rigorous evaluation, the dataset was partitioned into training, validation, and test subsets using a stratified 70:20:10 ratio, preserving the

DOI: 10.54254/2755-2721/2026.TJ30950

class distribution across all partitions. Recognizing the inherent risk of overfitting associated with a relatively small dataset, multiple data augmentation strategies were implemented during the training phase. These included random rotations within a 20-degree range and horizontal flipping, which simulate the varying orientations of leaves in natural environments and substantially enhance the model's generalization performance.

The selection of the model architecture was guided by the imperative of edge deployment. MobileNetV2 was chosen as the backbone due to its pioneering use of depthwise separable convolutions, which drastically reduce the computational complexity and parameter volume compared to conventional convolutional networks, without a commensurate sacrifice in accuracy [8]. This efficiency renders it ideally suited for the computational constraints of the Raspberry Pi. Transfer learning was adopted to leverage knowledge from the large-scale ImageNet dataset [7]. The pre-trained MobileNetV2 model was modified by removing its original classification head. A new custom classifier was appended, comprising a Global Average Pooling layer to reduce dimensionality, a Dense layer with 128 units and ReLU activation function for non-linear feature learning, and a final 3-unit Softmax output layer to perform class classification.

The model training process was executed in two distinct phases to balance training stability and predictive performance. Initially, the base MobileNetV2 layers were frozen, and only the weights of the newly appended classifier were trained over 10 epochs. This enabled the model to rapidly adapt its newly added layers to the feature space of plant diseases. Subsequently, the latter blocks of the base network were unfrozen, and the entire model was fine-tuned with a low learning rate (1e-5) for an additional 10 epochs. This second phase facilitated subtle adjustments to the pre-trained features, specializing them for the specific task of disease recognition and thereby enhancing final accuracy..

### 3.3. Experimental setup

The trained model was converted to TensorFlow Lite format and subjected to post-training dynamic range quantization to reduce its size and accelerate inference [9]. The hardware platform is a Raspberry Pi 4B integrated with an official CSI camera module. The software environment comprises Raspberry Pi OS (Legacy) with the TFLite Runtime and Picamera2 library pre-installed to facilitate model inference and camera operation.

## 4. Experimental evaluation and results

The system was evaluated based on model accuracy, size, and inference time. Accuracy metrics were derived from the test set. Inference time was measured as the average time taken for the Raspberry Pi to process a single image.

The quantized lightweight model retained high predictive accuracy despite a substantial reduction in storage size. The performance on the test set is summarized in Table 1.

Table 1. Model performance on test set

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Tomato Healthy | 0.994 | 0.991 | 0.992 | 654 |
| Tomato Early Blight | 0.973 | 0.981 | 0.977 | 542 |
| Tomato Late Blight | 0.984 | 0.972 | 0.978 | 588 |
| Overall Accuracy | | | 0.985 | 1784 |

DOI: 10.54254/2755-2721/2026.TJ30950

The quantized lightweight model demonstrated strong performance across all three classes, as detailed in Table 1. The overall accuracy reached 98.5%, with the 'Tomato Healthy' class achieving a near-perfect F1-Score of 0.992, indicating the model's high reliability in identifying non-infected leaves. For the disease classes, both Early Blight and Late Blight were identified with F1-Scores exceeding 0.97. The slightly higher recall for Early Blight suggests the model is particularly effective at finding all instances of this disease, minimizing missed detections. These results validate that the model maintains high accuracy after optimization and is well-suited for the practical application of disease screening.

The system's efficiency is highlighted in Table 2, comparing the edge-based approach with a typical cloud-based solution.

Table 2. Comparison between cloud and edge deployment

| Aspect | Cloud-based Solution | Proposed Edge Solution |
|---|---|---|
| Inference Latency | High(500ms-2000ms,network-dependent) | Low(450ms) |
| Data Privacy | Low(data leaves device for processing) | High(data processed locally) |
| Network Dependency | Required (constant,stable connection) | Not required |
| Model Size | Not a primary constraint(can be 100+MB) | 3.2MB |
| Operational Cost | Recurring(subscription fees,data egress cost) | One-time(hardware cost) |
| Deployment Scenario | Ideal for centralized,large-scale data analysis | Optimal for real-time,in-field diagnostics |

A direct comparison between the traditional cloud-centric approach and the proposed edge computing solution is presented in Table 2. The edge system demonstrates decisive advantages in key areas for agricultural field applications. Notably, it eliminates the critical reliance on network connectivity, a factor that is frequently unstable in remote rural agricultural environments. Through processing data locally on the Raspberry Pi, it ensures near-instantaneous inference with a latency of only 450 milliseconds, crucial for real-time feedback. Furthermore, this local processing paradigm inherently enhances data privacy as sensitive farm images never leave the premises. The combination of a minimal model size (3.2 MB) and a one-time hardware investment makes this system a cost-effective and practical solution for widespread deployment in precision agriculture.

## 5. Discussion

The experimental results substantiate the core thesis of this work: that a carefully designed edge AI system can serve as a viable and advantageous alternative to cloud-based solutions for real-time plant disease diagnosis. The high accuracy (98.5%) achieved by the quantized model, as detailed in Table 1, is particularly noteworthy. It demonstrates that modern model optimization techniques, specifically post-training dynamic range quantization, can dramatically reduce model size with a negligible impact on predictive performance [8]. This finding is critical for the edge computing domain, where storage and memory are at a premium.

The system's performance characteristics, summarized in Table 2, directly address the limitations of cloud-centric models. The inference latency of 450 ms, achieved entirely locally, eliminates the delays associated with data transmission to and from a remote server. This speed is crucial for practical, in-field use where immediate feedback is required. Furthermore, the local execution paradigm inherently guarantees data privacy, a key concern for agricultural stakeholders, as sensitive images never leave the physical device. The integration of a minimal 3.2 MB model and a one-time

DOI: 10.54254/2755-2721/2026.TJ30950

hardware cost establishes a compelling economic case for this system, which stands in stark contrast to the recurring subscription fees and data egress costs of cloud services.

However, it is crucial to contextualize these strong results within the study's limitations. The model's performance is intrinsically linked to the quality and scope of its training data. The PlantVillage dataset, while a valuable benchmark, consists of images captured under controlled, laboratory-like conditions. Consequently, the model's robustness to the myriad of challenges present in a real agricultural setting—such as complex backgrounds, occlusions, varying stages of disease progression, and different lighting conditions—remains an open question and a primary target for future work.

## 6. Conclusion

This study has demonstrated the practical viability of deploying a lightweight CNN for crop disease identification on a resource-constrained edge computing device, namely the Raspberry Pi. The developed system, centered on the MobileNetV2 backbone architecture, achieved a compelling trade-off between predictive performance and computational efficiency. Following training on a curated subset of the PlantVillage dataset targeting tomato health, early blight, and late blight, the model was subjected to post-training quantization. This critical optimization step reduced the model's disk footprint from approximately 12 MB to a mere 3.2 MB—representing a reduction of nearly 75%—thereby rendering it feasible for embedded edge deployment. The final evaluation on a separate test set yielded an overall accuracy of 98.5%, with the quantized model achieving an average inference time of 450 milliseconds on the Raspberry Pi 4B. These results culminate in a central conclusion: for time-sensitive and privacy-conscious agricultural applications, the edge computing paradigm offers a functionally superior and economically more accessible alternative to cloud-centric solutions. This implementation effectively translates the theoretical capabilities of deep learning into a tangible, low-cost tool with potential for direct in-field deployment.

While the study demonstrates high accuracy using the PlantVillage dataset, it faces limitations such as performance uncertainty in real farming conditions, a narrow focus on three categories within one crop, and a lack of adaptability to new data or disease variants post-deployment. To enhance agricultural relevance, the system's diagnostic capabilities need to be broadened. To address these limitations and build upon the present work, future research will be directed along several key avenues. The immediate next step involves constructing of a more robust dataset comprising images captured in diverse field settings, which is essential for training a model resilient to real-world variabilities. Concurrently, the exploration of more computationally efficient network architectures like EfficientNet-Lite or NASNet-Mobile has the potential to yield enhanced performance within the same computational budget constraints. Beyond optimizing the vision model, a promising direction is the development of a continuous learning framework that would allow the deployed system to incrementally learn from farmer feedback or new data without suffering from catastrophic forgetting of previous knowledge. Lastly, integrating this visual diagnosis system with heterogeneous data streams, such as IoT soil sensors or drone imagery, could pave the way for a holistic crop management system, moving from singular disease detection towards comprehensive plant health assessment.

## References

[1] FAO. (2021). The impact of disasters and crises on agriculture and food security: 2021. Food and Agriculture Organization of the United Nations.

DOI: 10.54254/2755-2721/2026.TJ30950

[2] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7, 1419.
[3] Li, L., Zhang, S., & Wang, B. (2021). Plant disease detection and classification by deep learning—A review. IEEE Access, 9, 56683-56698.
[4] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637-646.
[5] Satyanarayanan, M. (2017). The emergence of edge computing. Computer, 50(1), 30-39.
[6] Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161, 272-279.
[7] Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big Data, 3(1), 9.
[8] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).
[9] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., ... & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2704-2713).