Plant Disease Detection and Classification Using Machine Learning Models on a Jetson Nano

**Author:** Usama Mohiuddin (mohi0340@mylaurier.ca)

**Affiliation:** Wilfrid Laurier University Under the supervision of Professor Emad Mohammed

Abstract  
Agriculture plays a pivotal role in sustaining human civilization in India. Early detection of plant diseases is crucial to mitigate yield losses and ensure food security. This research explores the development, evaluation, and deployment of machine learning models on an edge device for efficient and real-time plant disease detection. Traditional approaches for diagnosing plant diseases are manual, subjective, and prone to errors. With advancements in machine learning (ML) and computer vision, these limitations can be overcome. However, real-world applications, especially in remote or resource-constrained environments, require lightweight, high-performance models optimized for edge devices. This research aims to compare the performance of multiple ML models—ResNet, MobileNet, EfficientNet, Inception and Custom CNN —for plant disease detection. The goal is to identify the best-performing models and deploy them on a Jetson Nano edge device integrated with a Yahboom robot for real-time agricultural use. Using the PlantVillage dataset, various models were trained and validated on plant leaf images, representing multiple diseases and healthy conditions. The models were optimized for inference efficiency and deployed using TensorFlow Lite. Performance metrics, including accuracy, F1 score, precision, and recall, were compared, followed by real-time testing on the Yahboom robot. EfficientNet outperformed other models with 98% accuracy, demonstrating superior classification capabilities. When deployed on the Jetson Nano, it achieved 95% real-time inference accuracy. MobileNet also performed well due to its lightweight architecture, making it ideal for edge deployment. This study demonstrates the potential of integrating ML with edge computing for real-time plant disease detection. Future work includes extending this approach to more complex datasets, deploying models on drones, and scaling applications to broader agricultural tasks.

**Note:** Make note of keywords.

# Introduction

## What do we know so far about the problem?

#### What is the problem? Define the problem that you have addressed in your work. Focus on the constraints that make the problem challenging.

The agricultural sector is the backbone of many economies, yet it remains vulnerable to numerous challenges, with plant diseases ranking among the most severe. Plant diseases are caused by pathogens such as fungi, bacteria, and viruses, and their outbreaks can devastate crops, leading to food shortages, price surges, and economic instability. Farmers often rely on manual inspection to detect diseases, a method that is not only labor-intensive but also inconsistent in accuracy. Compounding the problem is the difficulty of diagnosing diseases early. Initial symptoms may appear subtle, and the overlapping visual features of different diseases can easily confuse even experienced agronomists. These issues are exacerbated in regions with limited access to agricultural expertise, leading to delayed diagnoses and treatment. Real-world constraints make the problem even more challenging. The scale of modern farms, the variability in environmental conditions, and the diversity of crops necessitate solutions that can operate in dynamic and resource-constrained environments. This complexity demands robust, scalable, and efficient systems that outperform traditional diagnostic methods.

#### Why is it important? Here, you should illustrate the motivation behind finding a solution to this problem and not necessarily your own solution. What is the significance of solving this problem? What are the benefits we will save or gain, and what is the impact of the solution?

The importance of early and accurate plant disease detection cannot be overstated. Addressing this problem directly impacts global food security by preventing losses that could affect millions of people. Early detection ensures timely intervention, reducing the dependency on chemical pesticides and fertilizers, which often harm the environment and contribute to soil degradation. From an economic perspective, improved disease management reduces costs for farmers, enhances productivity, and ensures higher crop quality. Additionally, automated solutions can relieve the burden on farmers, allowing them to focus on other aspects of agriculture, such as soil health and crop diversification. Beyond economic and environmental benefits, solving this problem promotes equity in agriculture. Marginalized farmers in remote regions, often overlooked by traditional agricultural support systems, stand to benefit significantly from scalable and accessible technological solutions.

#### Are the current solutions not good enough? Why do we need your solution? What does your solution bring to the table?

While the introduction of machine learning and computer vision techniques has shown promise, most existing solutions fall short in real-world applications. Many ML models require high computational resources, making them unsuitable for deployment in resource-constrained environments. Cloud-based solutions introduce latency and depend heavily on stable internet connectivity, which is unavailable in many rural agricultural areas. On-device solutions face challenges of their own. Many edge computing platforms lack the processing power to run complex ML models efficiently. Additionally, current implementations often fail to account for real-time inference requirements, energy efficiency, and robustness against environmental variability. Our proposed solution bridges these gaps by optimizing ML models for deployment on edge devices like the Jetson Nano. These lightweight, high-performance models ensure real-time inference capabilities while maintaining accuracy. By integrating these models into an affordable and mobile platform such as the Yahboom robot, we demonstrate a practical solution for real-world agricultural applications.

# Literature Review

## **Concise, critical, and chronological discussion of the related works**

#### 2.1.1. What are the previous works related/similar to your work?

There has been research conducted on pretrained models on the PlantVillage Dataset, but they have not utilized putting that model onto an edge devices such as a Jetson Nano. Research has already explored CNN architectures like ResNet, EfficientNet, and MobileNet for plant disease detection. A study by Mohanty et al. used deep learning models trained on the PlantVillage dataset to achieve high classification accuracy for multiple plant diseases. Transfer learning approaches have been utilized by researchers to leverage pre-trained models such as ResNet and MobileNet for agricultural datasets. Edge computing applications have also been investigated, with researchers deploying TensorFlow Lite models on devices like Raspberry Pi for field-level disease detection.

#### 2.1.2. Clearly state the contribution of each work: “What did they do to solve the problem that you investigated?”

Previous works leveraged deep CNNs for high-accuracy plant disease classification. Some studies focused on transfer learning to enhance performance with limited data. Augmentation techniques were employed to address dataset imbalances.

#### 2.1.3. Clearly state the advantages and disadvantages of each work. You should be critical and rational. Clearly state your opinion on what is needed to improve the related works, “if possible,” and why. This should be an attempt to highlight the contribution of your work

**Advantages**

The advantages consisted of having a high accuracy on benchmark datasets. Also being of Effective use of augmentation and transfer learning techniques.

**Disadvantages**

Meanwhile there was a lack of focus on real time applications. There consisted Limited evaluation under dynamic environmental conditions and there was an absence of scalability for large-scale farms or diverse crops.

#### 2.1.4. Choose a chronological order, e.g., time of publication, solution significance, etc.

#### Use subsections “headings” to organize your literature review and make it easier to follow.

#### At the end of this section, you must stress what is missed. Why do we need it? And how will you solve this problem to achieve the missing parts? With what data? And what methods? Also, you need to stress the significance of your work: “What is so special/unique” about the work?

#### After this table, you must summarize the findings of the literature review. Clearly state what the current “research gap,” which part/component of the gap you are going to address, how, why it is needed compared to the literature, why your suggested solution is significant, etc. is.

# Material and Methods

## Data

#### 3.1.1. Give a complete description of the data with descriptive statistics. Illustrate the difficulties with the data, e.g., missing values, noise, inconsistency, etc.

The dataset used in this research, the PlantVillage dataset, is one of the most comprehensive publicly available datasets for plant disease detection. It comprises over 50,000 labeled images of plant leaves, representing 38 distinct categories of healthy and diseased conditions. The diversity of this dataset stems from its inclusion of commonly cultivated crops such as apples, potatoes, tomatoes, and peppers, each of which has multiple subcategories indicating specific diseases. Despite its extensive size and breadth, several challenges make working with this dataset non-trivial.

The first challenge lies in the class imbalance inherent to the dataset. While certain diseases are well-represented, others have a significantly smaller number of samples. This imbalance can lead to biases during model training, where the model becomes overly confident in predicting well-represented classes at the expense of underrepresented ones. Another key challenge is the similarity between some diseases, as different ailments often manifest with overlapping visual symptoms, making it difficult for even advanced models to distinguish between them accurately. Moreover, the environmental variability present in the images—stemming from differences in lighting, camera angles, and background conditions—introduces additional noise, requiring sophisticated preprocessing techniques to ensure consistent model performance.

To prepare this dataset for machine learning tasks, extensive preprocessing was carried out. Each image was resized to a fixed dimension of 224x224 pixels, ensuring compatibility across different machine learning architectures. Normalization of pixel values to a range of [0, 1] was performed to stabilize the gradient descent process during model training. To enhance the dataset's variability and help the models generalize to unseen conditions, several augmentation techniques were applied. These included random horizontal and vertical flips, rotations, brightness adjustments, and zooming. By simulating real-world variability, these augmentations played a critical role in improving model robustness.

In addition to these standard preprocessing steps, advanced feature engineering techniques were explored. For example, color histograms were computed to capture the unique color distributions associated with specific diseases. Similarly, local binary patterns (LBP) were used for texture analysis, which proved especially useful for differentiating diseases with subtle surface variations. Furthermore, image segmentation using OpenCV techniques was applied to isolate leaf regions from their backgrounds, ensuring that the models focused solely on the relevant portions of the images.

#### 3.1.2. Data preprocessing: in this section, explain your efforts to clean and prepare the data to suit your analytical methods. Also, you should state the implication of this step of the further processing, e.g., expected accuracy

Preprocessing involved resizing images to 224x224 pixels, normalizing pixel values, and applying augmentation techniques such as rotation, flipping, and zooming. These steps ensured model robustness and improved generalization.

#### 3.1.3. If you have a feature engineering step, you must clearly state why you need it and what background knowledge drives you to create more features. This step must be supported by a literature review. If you say that “it was based on experts’ opinion,” you must add references; otherwise, it may raise huge concerns during the review process.

## Methods

#### 3.2.1. What are the hypotheses behind your solution?

#### 3.2.2. What are the facts and/or established hypotheses that you are going to use to test your hypotheses?

#### 3.2.3. What do you want to conclude “proof” from your hypotheses and why?

#### 3.2.4. What are the different methods (the method must be explained in concrete steps, pseudo code, flow chart/diagrams, etc.) you will use to implement this work? What are special or unique about the methods, and how will they help you develop your solution? This is your contribution, and you need to discuss the associated novelty (What is the novelty of the method (what did you introduce, modify or add to the logic, procedure, computation, math, or architecture of the existing method? This should be compared to other methods you stated in the table at the end of the lit review to show significance.

#### 3.2.5. Briefly state the required experiments to test and validate the hypotheses and why.

The foundation of this research is built upon evaluating and comparing multiple machine learning models for plant disease detection. The hypothesis driving this work posits that EfficientNet, due to its compound scaling approach, will achieve superior accuracy without sacrificing computational efficiency, while MobileNet will demonstrate its utility as a lightweight model capable of real-time inference on edge devices.

To test these hypotheses, five distinct machine learning models—EfficientNet, MobileNet, ResNet, Inception, and a custom CNN—were trained on the processed dataset. Each model was evaluated using metrics such as accuracy, precision, recall, and F1 score. During training, the Adam optimizer with a learning rate of 0.001 and categorical cross-entropy loss was used. Training was conducted for 50 epochs, with early stopping to prevent overfitting. The models were then converted into TensorFlow Lite format to enable deployment on the Jetson Nano.

One of the novel contributions of this research is the integration of these models into a Yahboom robot equipped with a Jetson Nano edge device. This required meticulous optimization to balance model complexity and hardware constraints. For instance, quantization techniques were applied to reduce model size and improve inference speed, enabling real-time operation without compromising accuracy.

## Experimental Setup

#### 3.3.1. Describe each experiment regarding input training data, testing data, training method, and algorithm steps. Each experiment's assumptions must be listed, discussed and validated. Illustrate if this is a new or less common method.

# Results and Discussion

#### 4.1. The results and discussion section are significant to reflect your understanding of the results and how the results support your hypotheses/claims. You should report the results in figures and tables with a critical discussion on how the results may lead to a considerable improvement in solving the problem. Also, discuss how the limitations of the experiment affect the results.

#### 4.2. The figures and table must be self-explained. The reader should not read many sections of the paper to be able to understand and interpret any given table or figure. The table and figure caption must be descriptive and very well stated.

#### 4.3. You must concisely compare the advantages and disadvantages of other related works. Why do you think your method gave better results? Is it just the data? Is it something you add that offers a better understanding of the problem? What are the reasons that your problem is challenging to be solved? Explain what you did to overcome this difficulty

#### 4.4. Highlight the results that supported and unsupported your hypotheses or claims and clearly state why.

#### 4.5. How can we use the results in other applications? How would you know that your results are not inclusive? Did you overfit your classifiers? Can you recommend how to incorporate your results into a more extensive system?

Custom CNN Model

A graph of a training and validation accuracy

Description automatically generatedA graph with red and blue lines

Description automatically generatedA screenshot of a computer

Description automatically generated

Efficient Net

A graph with a line

Description automatically generatedA graph of training and validation loss

Description automatically generated

Inception

A graph with a line and a red line

Description automatically generatedA graph with a red line

Description automatically generated

ResNet

A graph of a training and validation accuracy

Description automatically generatedA graph of a graph showing the difference between training and validation

Description automatically generated

Yolo8

A comparison of graphs with different colored lines

Description automatically generated with medium confidence

A comparison of a graph

Description automatically generated

MobileNet

A graph with a red line

Description automatically generatedA graph with red lines and blue lines

Description automatically generated

The experiments revealed significant insights into the performance of the selected models. EfficientNet emerged as the most accurate model, achieving a validation accuracy of 98%. Its compound scaling approach, which optimally balances depth, width, and resolution, allowed it to outperform other models in detecting plant diseases with minimal misclassifications. When deployed on the Jetson Nano, EfficientNet maintained an impressive real-time inference accuracy of 95%, demonstrating its robustness in resource-constrained environments.

MobileNet, while slightly less accurate with a validation accuracy of 96%, excelled in terms of inference speed. On the Jetson Nano, it processed an average of 25 frames per second, making it the most suitable model for real-time applications. This performance is attributed to its lightweight architecture, which uses depthwise separable convolutions to reduce computational complexity.

ResNet, despite achieving a commendable accuracy of 94%, struggled with inference speed due to its heavier architecture. Inception, on the other hand, delivered moderate accuracy of 92% but faced challenges in maintaining efficiency during deployment. The custom CNN, while the least accurate at 90%, demonstrated the potential for creating lightweight architectures tailored to specific tasks.

The real-time performance of these models was evaluated using the Yahboom robot in a simulated agricultural environment. EfficientNet and MobileNet excelled in detecting diseases under varying lighting and background conditions, reaffirming their suitability for deployment in real-world scenarios. However, certain limitations were observed. The models occasionally struggled with diseases exhibiting overlapping visual symptoms, highlighting the need for further refinement in feature extraction techniques. Additionally, the Jetson Nano, while powerful, faced occasional latency issues when handling rapid frame processing with larger models like ResNet.

# Conclusion, Limitation and Future Work

#### 5.1. Briefly summarize the paper and state clearly, “What does the paper add? " This is your contribution; be careful! Your contribution must go beyond using existing blocks, simple lab experiments, restating facts, and/or an effort leading to **obvious** conclusions. It should be concise, rational, realistic, and solid, with an interesting conclusion that adds to our understanding of the problem.

This research demonstrates the potential of integrating machine learning with edge computing for real-time plant disease detection. By evaluating and deploying multiple machine learning models on the Jetson Nano, it provides a scalable and cost-effective solution for addressing a critical agricultural challenge. EfficientNet stood out as the most accurate model, while MobileNet balanced speed and accuracy effectively, making both models ideal for edge deployment.

Despite its success, this study has certain limitations. The PlantVillage dataset, while extensive, lacks the environmental diversity required for robust real-world applications. Diseases captured in controlled settings may not fully represent field conditions, where factors like occlusions, varying light intensities, and background clutter play a significant role. Moreover, the Jetson Nano, while suitable for lightweight models, struggles to accommodate larger architectures, limiting its scalability for more complex tasks.

Future work aims to address these limitations by collecting real-world data from diverse agricultural settings to enhance model robustness. Further, exploring advanced optimization techniques such as model pruning and quantization will enable the deployment of more complex models without compromising performance. Scaling this approach to broader agricultural tasks, such as pest detection and soil quality assessment, will expand its applicability. Additionally, deploying these models on drones for aerial monitoring will allow for larger-scale applications, transforming the way precision agriculture is practiced.

#### 5.2. INTERESTING USAGE SCENARIOS (how to generalize the results to other use cases in other fields)

#### 5.3. What are the limitations of your results compared to others?

#### 5.4. The future work section should contain a feasible rational solution to the work limitation you want to pursue afterwards.

# References

<https://openagriculturejournal.com/VOLUME/18/ELOCATOR/e18743315321139/FULLTEXT/>

<https://www.kaggle.com/c/plant-pathology-2020-fgvc7/data>