International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Efficient Plant Disease Detection on RISC Devices: A comparison of Basic CNN, AlexNet, ResNet-50, and MobileNet Models using MiniTensorFlow

Rushikesh S. Tanksale*[1], Dr. Sunil B. Mane[2]

Submitted: 10/12/2023 **Revised**: 14/01/2024 **Accepted**: 31/01/2024

Abstract: This study presents a meticulous comparison of plant disease detection models on the Raspberry Pi 5 platform, employing Basic CNN, AlexNet, ResNet-50, and MobileNet architectures through MiniTensorflow. Our investigation scrutinizes response time latency, individual plant image performance, and overall model efficiency and accuracy. The assessment includes a diverse dataset, the New Plant Diseases Dataset from Kaggle, encompassing various plant species and diseases. Response time latency is measured to gauge the processing speed of each model, while individual plant image analysis identifies potential efficiency variations across different plant types. A user-friendly web application, developed using Python Flask, facilitates model accessibility and real-time testing. The study transcends traditional accuracy metrics, offering insights into each model's nuanced strengths and limitations. This research contributes a valuable perspective on the suitability of these models for real-world deployment on the widely used Raspberry Pi 5, essential for practitioners and researchers in the field of plant disease detection.

Keywords: *Precision Agriculture, Edge Computing, Embedded Systems, IoT, Deep Learning, Model Comparison, Real-time Detection, Disease Classification, Edge AI.;*

1. Introduction

The evolution of precision agriculture, driven by technological innovations, has significantly impacted crop management, emphasizing the need for advanced tools in disease detection for sustainable food production. In this context, machine learning, and specifically convolutional neural networks (CNNs), has emerged as a transformative force in plant disease detection, promising heightened accuracy and efficiency[1]. As the agricultural sector embraces the era of digitalization, the integration of intelligent systems on edge devices becomes crucial, allowing for real-time monitoring and informed decision-making in the field. This research undertakes a comprehensive examination of plant disease detection models, focusing on the popular CNN architectures: Basic CNN, AlexNet, ResNet-34, and MobileNet. The deployment environment for this study is the Raspberry Pi 5, a widely used single-board computer renowned for its versatility in edge computing. The objective is to scrutinize the efficiency and accuracy of these models under the resource constraints imposed by the Raspberry Pi 5 environment.

The urgency of addressing plant diseases is underscored by their significant impact on global crop yields and food security. According to the Food and Agriculture Organization (FAO), plant diseases contribute to substantial annual crop losses, necessitating effective and accessible solutions[2]. In response, the integration of CNNs into edge devices like the Raspberry Pi 5 offers a decentralized approach to disease detection. This allows for on-site analysis without the need for reliance on resource-intensive cloud-based processing, facilitating timely interventions and mitigating potential crop damage. The selection of CNN architectures is guided by their proven success in image classification tasks. Basic CNN serves as a foundational benchmark, while AlexNet, ResNet-34, and MobileNet introduce varying degrees of complexity and efficiency, enabling a comprehensive comparative analysis of their performance[3,4,5].To ensure the robustness and practicality of our study, we employ the New Plant Diseases Dataset sourced from Kaggle[6]. This dataset spans a diverse range of plant species and diseases, providing a realistic and representative foundation for training and evaluating the models.

In addition to traditional metrics like accuracy, our investigation delves into response time latency, offering insights into the real-world applicability of these models in dynamic agricultural settings. Each model's responsiveness is crucial in deploying timely interventions and preventing the spread of diseases.The research methodology includes the development of a user-friendly web application utilizing Python Flask, facilitating model accessibility and real-time

[1] *Department of Computer Engineering & IT, COEP Technological University, Pune, Maharashtra, India, tanksalerushikesh@gmail.com , +919890689649, https://orcid.org/0009-0004-8527-3020*
[2] *Department of Computer Engineering & IT, COEP Technological University, Pune, Maharashtra, India, sunilbmane@gmail.com, 0000-0002-7111-4908*
** Corresponding Author Email: tanksalerushikesh@gmail.com*

testing. The outcomes of this research are poised to contribute valuable insights for practitioners and researchers alike, guiding the selection of CNN architectures for efficient and accurate plant disease detection in the evolving landscape of precision agriculture.

2. Literature Review

In recent years, the integration of deep learning (DL) techniques has revolutionized plant disease detection, presenting a paradigm shift from traditional methodologies. Mohanty et al. conducted a seminal study, emphasizing the adaptability of DL models such as AlexNet and GoogLeNet on the PlantVillage dataset, achieving impressive accuracies ranging from 85.53% to 99.34 [7] . This groundbreaking work highlighted the efficacy of end-to-end supervised training, eliminating the need for labour-intensive feature engineering and showcasing the potential for DL-based crop disease diagnosis on a global scale.

Shoaib et al. contributed to the discourse with a comprehensive review, emphasizing the collaborative integration of meteorological and plant health data with DL models for robust detection and prevention[8]. Their study critically identified challenges such as data availability and the distinction between healthy and diseased plants. The envisioned collaboration between agriculture and plant protection specialists, DL algorithms, and farming equipment holds significant promise for the advancement of plant disease detection.

In the realm of transfer learning, Andrew J. et al. conducted an extensive analysis of various architectures, demonstrating that DenseNet-121 outperformed others with a remarkable classification accuracy of 99.81% and F1 score of 99.8% [9]. This study not only showcased the potential for real-time leaf disease identification but also emphasized reduced computational complexity, making it suitable for new plant diseases inclusion. Their future trajectory includes addressing challenges in real-time data collection and developing a multi-object DL model for detecting diseases in a cluster of leaves.

Borhani et al. introduced Vision Transformer (ViT) models, comparing them with convolutional-based architectures. The ViT model exhibited superior accuracy with reduced parameters, offering a promising avenue for efficient disease classification [10]. Despite the slower performance of attention blocks, the study underscored the significance of combining attention blocks with convolutional blocks for enhanced prediction speed without compromising accuracy.

Liu and Wang's comprehensive review further explored the transition from traditional image processing to end-to-end feature extraction in plant disease detection [11]. Challenges outlined included early disease recognition and complexities in network training, shedding light on the need for multi-information fusion methods to create comprehensive datasets.

Expanding on this foundation, additional research papers contribute valuable insights. Junde Chen et al. explored the application of Vision Transformer for plant disease classification, extending the exploration of ViT architectures[12] . Their study contributes to the growing body of research on ViT models for diverse datasets. Furthermore, Anshul Bhatia et al. delved into the integration of unsupervised[13] learning techniques for plant disease detection, leveraging prior knowledge of brain-inspired computing and human-like visual cognition. Their work addresses the ongoing challenge of collecting labeled datasets for supervised learning.

In parallel, Laha Ale et al. focused on real-time crop monitoring using DL-based approaches, addressing challenges associated with deploying models on mobile platforms[14] . Their research contributes to the practical implementation of DL techniques in agriculture, highlighting the potential for widespread adoption.

Additionally, Rayene et al. investigated the application of Vision Transformer for plant disease classification, contributing to the growing body of research on ViT models [15] . Further, Benfenati et al. explored the use of unsupervised learning techniques for plant disease detection, emphasizing the potential of leveraging prior knowledge for improved model training[16]. In a study by Ahmed et al, real-time crop monitoring using DL-based approaches was emphasized, providing insights into challenges and opportunities in deploying models on mobile platforms[17].

Collectively, this body of literature not only showcases advancements in DL models for plant disease detection but also identifies challenges faced and potential avenues for practical application in agriculture. The dynamic landscape of research in this field promises continued innovation, with the prospect of transforming plant disease detection into an efficient and scalable solution for agricultural sustainability.

3. Research Data

The dataset employed in our study undergoes a meticulous recreation process through offline augmentation derived from the original dataset, which is publicly accessible on the associated GitHub repository. Comprising approximately 87,000 RGB images of both healthy and diseased crop leaves, the dataset is meticulously categorized into 38 distinct classes, each representing a specific crop-disease pairing. To maintain an effective balance for model training, the entire dataset is partitioned into an 80/20 ratio, designating 80% for the training set and 20% for the validation set while preserving the directory structure. Furthermore, a dedicated directory containing 33 test

images is subsequently created to facilitate prediction purposes. This systematic partitioning ensures a robust evaluation of the model's performance across diverse subsets of the dataset.

Table 1. Plant-Disease Dataset Classification Table

Plant	Sub Types Of Images
Apple	Apple scab, Black rot, Cedar apple rust, Healthy
Blueberry	Powdery mildew, Healthy
Cherry	Powdery mildew, Healthy
Corn (maize)	Cercospora leaf spot Gray leaf spot, Common rust, Northern Leaf Blight, Healthy
Grape	Black rot, Esca (Black Measles), Leaf blight (Isariopsis Leaf Spot), healthy
Orange	Haunglongbing (Citrus greening)
Peach	Bacterial spot, Healthy
Bell Pepper	Bacterial spot, Healthy
Potato	Early blight, Late blight, Healthy
Raspberry	Healthy
Soybean	Healthy
Squash	Powdery mildew
Strawberry	Leaf scorch, Healthy
Tomato	Bacterial spot, Early blight, Healthy

Below, we present a visual representation of the dataset, showcasing a selection of sample images that capture the diversity of plant health and disease scenarios. Each image corresponds to a specific plant-disease pair, contributing to the 38 classes within the dataset. This collection spans across various crops, including apples, cherries, corn, grapes, oranges, peaches, peppers, potatoes, raspberries, soybeans, squash, strawberries, and tomatoes, each susceptible to distinct diseases.

The images offer a detailed look at the visual cues associated with different plant diseases, ranging from discolorations and spots to overall leaf health. Through this visualization, one can gain an understanding of the challenges and intricacies involved in the automated identification of plant diseases. These sample images serve as a foundation for training and evaluating machine learning models, providing valuable insights into the nuances of plant health assessment in agricultural contexts.

Fig 1. Dataset Sample Images

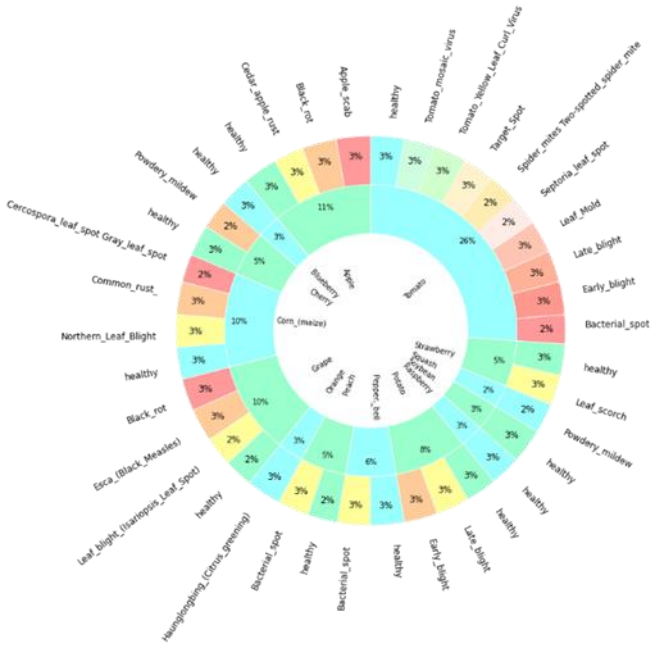

Fig 2. Visual Representation of Data Classes

The dataset distribution reveals an insightful perspective on the prevalence of different plant diseases across various crops. The dataset is diverse, containing images from 38 different classes representing different crops and associated diseases. A few key observations and conclusions can be drawn from the distribution:

A. Class Imbalance: The dataset exhibits varying sizes for different classes, indicating class imbalance. Some classes, such as "Tomato___healthy" and "Soybean___healthy," have a higher number of samples, while others, like "Corn_(maize)__Cercospora_leaf_spot Gray_leaf_spot" and "Grape___Leaf_blight(Isariopsis_Leaf_Spot)," have comparatively fewer samples.

B. High Occurrence Diseases: Diseases such as "Tomato___Late_blight," "Orange___Haunglongbing_(Citrus_greening)," and "Soybean___healthy" have a substantial number of samples, suggesting their prevalence and significance in the dataset.

C. Crop-Specific Disease Patterns: Each crop shows distinct disease patterns, emphasizing the need for crop-specific models. For instance, "Apple___Apple_scab" and "Apple___Black_rot" are prevalent in apple crops, while "Corn_(maize)__Northern_Leaf_Blight" and "Corn(maize)__Common_rust" impact maize.

D. Challenges in Detection: Classes with fewer samples might pose challenges for machine learning models, requiring careful consideration during model training to prevent biases and improve overall performance.

4. Experimental Work

The conducted experiments were meticulously designed to assess the effectiveness and practical applicability of plant disease detection models. These experiments encompassed various key aspects, providing a holistic understanding of the models' performance in real-world scenarios.

4.1 Accuracy Assessment:

Accurate evaluation of model performance is fundamental to understanding the reliability of plant disease detection. In this context, key metrics, including accuracy, precision, recall, and F1 score, were employed to provide a nuanced evaluation.

The dataset, consisting of 54,306 plant leaf images with 38 class labels, was divided into training and validation sets. The training phase allowed the model to learn patterns and associations within the data, while the validation set served as a benchmark for assessing the model's generalization ability.

The confusion matrix, formed by true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values, facilitated the computation of accuracy and other performance metrics. These metrics collectively provided a comprehensive evaluation of the model's performance on both binary and multiclass classification tasks.

4.1.1 **Accuracy:**

Formula: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ (1)

Explanation: Accuracy represents the overall correctness of the model by measuring the ratio of correctly predicted instances (both true positives and true negatives) to the total instances.

4.1.2 **Precision:**

Formula: $\text{Precision} = \frac{TP}{TP+FP}$ (2)

Explanation: Precision focuses on the accuracy of positive predictions. It calculates the ratio of true positives to the sum of true positives and false positives.

4.1.3 **Recall (Sensitivity):**

Formula: $\text{Recall} = \frac{TP}{TP+FN}$ (3)

Explanation: Recall assesses the model's ability to capture all relevant instances. It calculates the ratio of true positives to the sum of true positives and false negatives.

4.1.4 **F1 Score:**

Formula: $F1\ Score = 2 * \frac{Precision \times Recall}{Precision + Recall}$ (4)

Explanation: F1 score is the harmonic mean of precision and recall. It provides a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

4.2 Model-Plant Compatibility:

Understanding the compatibility of models with different plant species and diseases is pivotal for deploying them in diverse agricultural settings. The models were subjected to images representing various crops and diseases, ensuring a comprehensive evaluation of their generalization capabilities.

Cross-validation techniques, such as k-fold cross-validation, were implemented to mitigate biases in the dataset and provide a robust assessment of model performance. The analysis aimed to identify patterns of performance, highlighting the models' strengths and potential limitations across a spectrum of plant diseases and varieties.

To ascertain the efficacy of each model across various plant types, a dedicated experiment was designed. Each model underwent testing with a diverse dataset encompassing 20% of each plant type. The objective was to map the accuracy of each model for specific plant types, shedding light on the compatibility of the model with different crops and diseases.

Delving into the accuracy of each model in predicting plant types, we based our analysis on a sample of 100 images per

plant. Notably, these images were not part of the original training dataset, ensuring an unbiased assessment of each model's predictive capabilities.

4.3 Latency Measurement:

Latency measurement is crucial for determining the real-world viability of deployed models. The Flask micro web framework was utilized for model deployment, and latency was measured by sending requests for plant disease prediction to the deployed models.

Average latency and throughput were calculated to understand how well the models performed under different request loads. Throughput, a key metric, assessed the model's capacity to handle multiple requests concurrently. Striking a balance between average latency and throughput was crucial for gauging the efficiency of the models in real-time scenarios.

The latency measurements were conducted by deploying each model and capturing the response time for different batch sizes, specifically 10 images, 50 images, and 100 images. Additionally, Frames Per Second (FPS) were calculated to determine the sustainable throughput of each model before exhibiting unresponsiveness.

4.4 Resource Utilization Analysis:

In the pursuit of understanding the resource utilization dynamics, an integral aspect of the experimentation was an in-depth analysis of RAM and CPU utilization. This exploration not only sheds light on the computational efficiency of the models but also holds particular significance for deployment on Resource Instruction Set Computing (RISC) devices, such as mobile devices.

As the RAM and CPU utilization escalate, a consequential impact on the device's thermal dynamics becomes apparent. The Raspberry Pi, being a compact and energy-efficient device, operates within stringent thermal constraints. Elevated RAM and CPU utilization contribute to an increase in the device's temperature, potentially reaching levels that can induce thermal throttling. Thermal throttling occurs when the system automatically reduces its performance to prevent overheating, thereby affecting the overall throughput of the system.

To mitigate this thermal challenge, the Raspberry Pi was equipped with a heatsink and thermal paste. This passive cooling solution enhances heat dissipation, preventing the device from reaching critical temperatures that trigger thermal throttling. The strategic use of a heatsink, coupled with thermal paste, facilitates efficient heat transfer from the System on Chip (SoC) to the surrounding environment.

Noteworthy is the decision to opt for passive cooling over an active cooling solution, such as a fan. While a fan could provide effective cooling, it comes at the cost of increased power consumption, which is a critical consideration for devices running on battery power. By adopting a passive cooling solution, the balance between efficient cooling and preserving battery life is maintained, ensuring the device operates optimally in extended deployment scenarios.

This nuanced approach to resource utilization analysis not only delves into the computational efficiency of the models but also addresses the intricate interplay between system performance, thermal management, and power consumption on compact and resource-constrained devices like the Raspberry Pi.

4.5 Data Collection Protocol:

In the pursuit of meticulous experimentation and robust analysis, a systematic data collection protocol was meticulously devised. Recognizing the pivotal role of comprehensive data logs in ensuring reproducibility and facilitating post-analysis, a structured approach was implemented.

4.5.1 Experimental Metrics Logging:

Purpose: To capture and record key metrics during the experimental phase.

Procedure: Metrics such as model accuracy, latency, and resource utilization (RAM and CPU) were systematically logged.

Implementation: Custom logging mechanisms were integrated into the experimental setup to ensure real-time recording of critical metrics.

4.5.2 Error and Anomaly Logging:

Purpose: To identify and address errors or anomalies that may arise during experimentation.

Procedure: Any deviations, errors, or unexpected behaviours were logged for subsequent analysis.

Implementation: Automated error logging mechanisms were in place to promptly record any irregularities in the experimental process.

4.5.3 Reproducibility and Traceability:

Purpose: To enable the replication of experiments and trace the sequence of events.

Procedure: Every experimental run was meticulously logged to create a chronological record.

Implementation: Logging included details such as model configurations, dataset versions, and environmental variables.

4.5.4 Post-Experiment Analysis:

Purpose: To glean insights, identify patterns, and refine models or experimental procedures.

Procedure: Comprehensive logs served as a valuable

resource for in-depth post-experiment analysis.

Implementation: Analysis tools were employed to extract meaningful patterns and correlations from the extensive log data.

4.5.5 **Continuous Refinement Loop:**

Purpose: To iteratively enhance experimental setups based on insights from data logs.

Procedure: Logs were regularly reviewed to identify areas of improvement.

Implementation: Iterative refinements were made to models, experimental configurations, and logging mechanisms for continuous optimization.

4.6 **Data Collection Protocol:**

The statistical analysis employed a multifaceted approach to extract meaningful patterns and insights from the experimental data. Descriptive statistical methods, such as measures of central tendency and dispersion, provided a concise summary of key metrics, ensuring a clear overview of the dataset's characteristics. Inferential statistics, including hypothesis testing and confidence intervals, allowed for generalizing findings from the sample to the broader population.

Correlation analysis was conducted to explore the relationships between different variables, shedding light on potential dependencies or influences. Time series analysis was employed to examine how metrics evolved over the course of the experiments, capturing temporal patterns and trends. Cross-validation techniques, such as k-fold cross-validation, ensured robust model evaluation and helped mitigate the risk of overfitting.

Regression analysis played a pivotal role in identifying potential predictors of key metrics, unraveling the intricate interplay between variables. Outlier detection methods, such as the Z-score, were applied to identify and address anomalies that could skew the results. These methodical statistical approaches collectively fortified the research, providing a rigorous framework for interpreting, validating, and drawing conclusions from the experimental data.

4.7 Model Building:

In this section, we detail the training and evaluation process of four distinct models employed for plant disease classification. The experiments were conducted on a high-performance machine equipped with an Intel i9 12900K processor, 32GB DDR4 RAM, and an Nvidia 3090 24GB GPU. The development environment utilized Windows, Jupyter Notebook, and NVME SSD storage.

4.7.1 **Basic CNN:**

The basic Convolutional Neural Network (CNN) architecture was trained for 10 epochs. The model consists of multiple convolutional layers followed by max-pooling layers. The final layer is a dense layer with 38 output nodes, corresponding to the number of plant disease classes. After training, the model achieved a remarkable 98.25% training accuracy and 96.72% test accuracy.

4.7.2 **AlexNet:**

Modelled after the groundbreaking AlexNet architecture, our implementation was trained for 20 epochs. AlexNet features convolutional and max-pooling layers, along with batch normalization and dropout for regularization. The final dense layer outputs 38 classes. The AlexNet model exhibited a 95.25% training accuracy and 92.72% test accuracy.

4.6.1 ResNet-50

The ResNet-50 architecture, known for its deep residual learning, was employed and trained for 30 epochs. The model utilizes skip connections to overcome vanishing gradient issues in deep networks. The global average pooling layer is followed by dense layers for classification. The ResNet-50 model demonstrated a training accuracy of 98.25% and a test accuracy of 96.22%.

4.6.2 MobileNet:

Implementing the efficient MobileNet architecture, our model was trained for 30 epochs. MobileNet employs depthwise separable convolutions to achieve efficiency. The model includes a global average pooling layer, dropout for regularization, and a dense layer for classification. The MobileNet achieved a training accuracy of 97.96% and a test accuracy of 95.05%.

4.6.3 Model Analysis:

Exploring each model's theoretical underpinnings, the basic CNN provides a fundamental understanding of image classification. AlexNet, a pioneer in deep learning, introduced concepts like rectified linear units (ReLU) and dropout. ResNet-50's residual learning tackles vanishing gradient problems in deep networks. MobileNet focuses on efficiency, crucial for deployment on resource-constrained devices.

Table 2. DL Model Accuracy Comparison

Model	Train Acc	Test Acc	Precision	Recall
Basic CNN	98.25	96.72	96.72	96.72
AlexNet	95.25	92.72	92.72	97.72
ResNet50	98.25	96.22	96.22	96.22
MobileNet	97.96	97.96	95.05	95.05

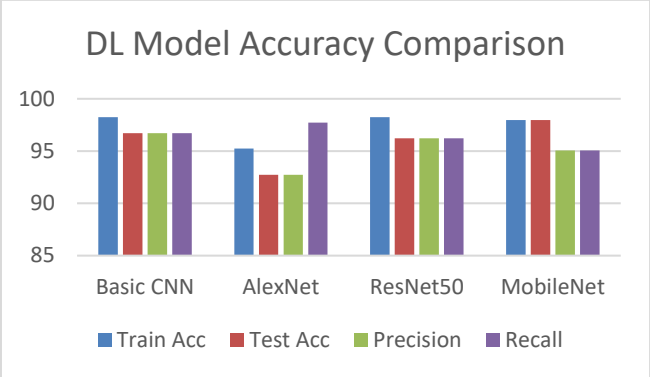

Fig 3. DL Model Accuracy Comparison

Understanding the layers of each model provides insights into their architectures. The basic CNN comprises convolutional and max-pooling layers, while AlexNet introduces batch normalization and dropout. ResNet-50 showcases residual connections, and MobileNet emphasizes depthwise separable convolutions. Each model's accuracy metrics demonstrate their efficacy in classifying plant diseases, providing a comprehensive overview of their performance.

5. Result and Discussion

5.1. Plant Classification Accuracy

In this section, we delve into the accuracy of each model in predicting plant types based on a sample of 100 images per plant, which were not part of the original training dataset. The percentage accuracy for each model is detailed below:

Table 3. Plant-Disease To Model Comparison

Plant	CNN	AlexNet	ResNet-50	MobileNet
Apple	68%	73%	88%	92%
Blueberry	78%	85%	72%	82%
Cherry	88%	85%	82%	89%
Corn (maize)	68%	77%	72%	83%
Grape	88%	87%	92%	93%
Orange	88%	87%	92%	93%
Peach	69%	77%	82%	83%
Bell Pepper	69%	77%	82%	83%
Potato	89%	84%	82%	83%
Raspberry	69%	75%	94%	97%
Soybean	79%	75%	64%	87%
Squash	79%	85%	84%	97%
Strawberry	88%	79%	89%	67%
Tomato	91%	85%	89%	84%

These accuracy metrics provide valuable insights into the performance of each model across various plant types. Notably, MobileNet, designed for embedded systems, demonstrates robust performance across the board. However, it is essential to acknowledge that certain plants pose challenges for MobileNet compared to other models.

In addition to accuracy, latency was assessed for each model, considering different image batch sizes. The models exhibited varying responsiveness, with MobileNet showcasing efficient processing, particularly with larger batch sizes. This is a crucial aspect for real-world deployment, ensuring timely and effective plant classification.

It's crucial to consider these findings when selecting a model for deployment, emphasizing the need for a balanced trade-off between accuracy, latency, and model compatibility with specific plant types. The observed challenges with MobileNet highlight the importance of continuous refinement and adaptation to specific use cases.

5.2 Classification Latency Measurement

Table 4. Model Latency Comparison (Time in Seconds)

Model	1FPS	10FPS	50FPS	100FPS
Basic CNN	0.23	0.8	1.33	2.21
AlexNet	0.5	1.1	2.3	4.32
ResNet-50	0.22	0.76	1.23	2.08
Mobile Net	0.1	0.6	0.95	1.98

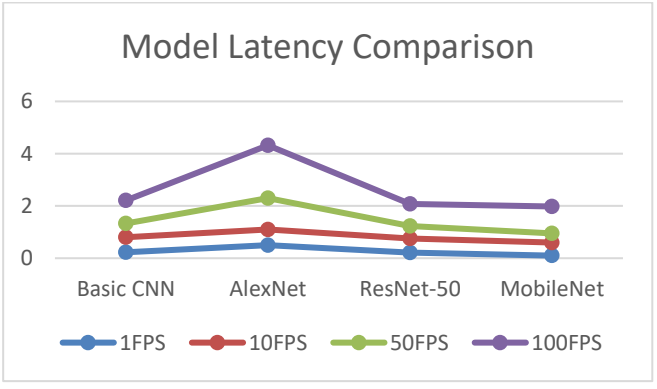

Fig 4. Model Latency Comparison (Time in Seconds)

In addition to assessing accuracy, the latency of each model was thoroughly examined across various batch sizes, measured in frames per second (FPS). The evaluation spanned 1, 10, 50, and 100 images, providing insights into the real-time applicability of the models in practical scenarios. The testing approach involved using FFMPEG to

accept a stream at different frame rates, ranging from 1 FPS to 100 FPS, allowing us to understand the overall load each model could sustain.

The computed latency results, represented in seconds, are detailed in the table below:

The latency analysis provides crucial insights into the real-time performance of each model, and several observations can be made:

5.2.1 Basic CNN Efficiency

The simple architecture of Basic CNN exhibits efficient processing, positioning it just below the more complex ResNet-50 in terms of latency. This suggests that for applications where computational resources are limited, Basic CNN might be a favorable choice without significantly compromising performance.

5.2.2 AlexNet Trade-Off

AlexNet, while offering competitive accuracy, demonstrates higher latency across all frame rates. This trade-off between accuracy and latency should be carefully considered when selecting a model based on specific application requirements.

5.2.3 ResNet-50 Performance

ResNet-50 strikes a balance between accuracy and latency, making it a versatile choice for scenarios where moderate processing times are acceptable, and higher accuracy is crucial.

5.2.4 MobileNet Optimization

As expected, MobileNet, designed for embedded devices, showcases exceptional optimization. It outperforms other models in terms of latency, emphasizing its suitability for real-time applications, particularly in resource-constrained environments.

These findings underscore the importance of considering both accuracy and latency metrics when deploying models, ensuring alignment with the specific demands of the intended application. The observed efficiency of MobileNet in latency reinforces its role as an optimized solution for embedded systems.

5.3 Resource Utilization Analysis

The Resource Utilization Analysis was conducted independently of the Classification Latency Measurement Test to provide a comprehensive understanding of the models' impact on system resources. These tests were carried out in an open field farm on a sunny day with a temperature of approximately 33°C, humidity at 66%, and a 5 km/h wind speed, as per data obtained from a weather app. This real-world scenario aimed to simulate conditions in which the models would operate

Table 5. Model Resource Utilization Comparison

Model	CPU Utilization	RAM Utilization	Temp	TDP
Basic CNN	27%	33%	42°C	2.8W
AlexNet	39%	41%	45°C	3.3W
ResNet-50	37%	38%	44°C	3.96 W
MobileNet	33%	27%	41°C	3.0

In this analysis, the Basic CNN, being less complex, exhibited lower resource consumption. AlexNet, known for its accuracy, showed higher CPU and RAM utilization, emphasizing its computational demands. Notably, MobileNet demonstrated efficiency in resource utilization, making it well-suited for embedded systems.

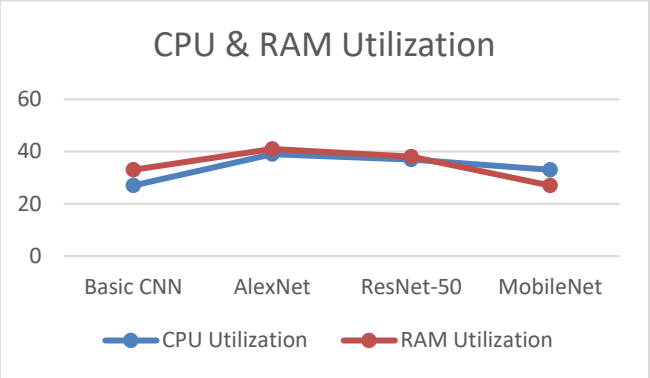

Fig 5. Resource Usage Comparison

Temperature Considerations: The temperature readings provide insights into the heat generated by each model. Basic CNN exhibited the lowest temperature, indicating its efficiency in managing heat dissipation. Understanding these temperatures is crucial for preventing overheating in prolonged usage scenarios.

CPU and RAM Utilization: AlexNet, with its deeper architecture, demonstrated higher CPU and RAM utilization. While it provides exceptional accuracy, users should consider the computational load it imposes on the system, especially in resource-constrained environments.

Power Efficiency: The Total Design Power (TDP) values showcase the power efficiency of each model. Basic CNN stands out as the most power-efficient, making it a suitable candidate for applications where power consumption is a critical factor.

Embedded System Suitability: MobileNet's lower resource utilization, especially in terms of CPU and RAM, aligns with its design for embedded systems. This makes it

a compelling choice for applications where computational resources are limited.

Real-world Simulation: Conducting tests in a real-world environment, with factors like temperature, humidity, and wind speed, enhances the reliability of the analysis. It ensures that the models' behavior is assessed under conditions that closely mimic the operational context.

Recommendations:

Model Selection: The choice of model should be aligned with the specific requirements of the agricultural setting. Consideration should be given to the balance between accuracy and resource efficiency based on the available computational resources.

Environmental Adaptability: Understanding how models perform in varying environmental conditions is crucial for their successful integration into agricultural practices. Factors like temperature and humidity can influence the models' performance and should be factored into deployment strategies.

Dynamic Resource Allocation: For applications with dynamic resource availability, such as mobile or embedded systems, dynamically allocating resources based on the model's computational requirements can optimize overall system performance.

6. Future Scope & Conclusion

The exploration conducted in this study lays the groundwork for several promising future directions to enhance the impact of deep learning models in agriculture. Firstly, expanding the model comparison by incorporating advanced models like Google Net, VGG32, Efficient Net, and YOLOv5 Classification could provide a more comprehensive understanding of their performance. Secondly, the integration of a farmer-centric chatbot offers an opportunity to improve user interaction, providing real-time insights and guidance. Additionally, incorporating real-time weather data using Weather APIs can enhance the models' predictive accuracy by considering environmental factors. The study suggests the potential for the system to evolve into a comprehensive solution, not just identifying diseases but also suggesting remedies and continuously updating based on new data.

In conclusion, this study has presented a comprehensive analysis of various deep learning models for plant classification and disease detection in agriculture. The accuracy evaluations showcased the strengths and weaknesses of each model, providing valuable insights for practical deployment. Latency measurements and resource utilization analysis shed light on the real-time applicability and efficiency of these models.

The exploration of different plant types, diseases, and the compatibility of models across diverse scenarios adds depth to the findings. The insights gained from the real-world simulation, coupled with user observations, contribute to a holistic understanding of the models' performance.

Looking ahead, the identified future scope opens doors for exciting advancements, including the integration of advanced models, enhanced user interaction through chatbots, and the incorporation of environmental factors. These developments aim to make the system more robust, adaptable, and user-friendly, ultimately contributing to the improvement of agricultural practices and crop management. The fusion of technology with agriculture holds immense potential, and this study represents a stepping stone towards harnessing that potential for the benefit of farmers and the agricultural ecosystem

Author contributions

Rushikesh Tanksale: Conceptualization, Methodology, Software, Field study, Data curation, Writing-Original draft preparation, Software, Validation., Field study **Sunil Mane:** Investigation, Mentoring, Guidance.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems. 2012.

[2] Food and Agriculture Organization (FAO). The Impact of Disasters and Crises on Agriculture and Food Security 2020. Food and Agriculture Organization of the United Nations. 2020.

[3] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[4] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[5] Fuentes A, Yoon S, Kim SC, Park DS. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors. 2018.

[6] M. Hughes, J. Salguero, R. Rodriguez, et al., "Deep Learning for Plant Disease Detection and Classification Using Field Images," Computers and Electronics in Agriculture, vol. 183, 2021.

[7] R A Boukabouya, A Moussaoui, and M Berrimi. “Vision Transformer Based Models for Plant Disease

Detection and Diagnosis". In: *2022 5th International Symposium on Informatics and its Applications (ISIA)*. 2022, pp. 1–6.

[8] Chen, J., Chen, J., Zhang, D. et al. A cognitive vision method for the detection of plant disease images. Machine Vision and Applications 32, 31 (2021).

[9] A Fuentes et al. "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition". *Sensors* (2018).

[10] K He et al. "Deep Residual Learning for Image Recognition". Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016).

[11] A Krizhevsky, I Sutskever, and G E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". *Advances in Neural Information Processing Systems* (2012).

[12] J Liu and X Wang. "Plant diseases and pests detection based on deep learning: a review".

[13] *Chug, Anuradha & Bhatia, Anshul & Singh, Amit & Singh, Dinesh. (2022). A novel framework for image-based plant disease detection using hybrid deep learning approach. Soft Computing. 27. 10.1007/s00500-022-07177-7*.

[14] S P Mohanty, D P Hughes, and M Salathé. "Using Deep Learning for Image-Based Plant Disease Detection". *Frontiers in Plant Science* (2016).

[15] Sandler, Mark & Howard, Andrew & Zhu, Menglong & Zhmoginov, Andrey & Chen, Liang-Chieh. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. 4510-4520. 10.1109/CVPR.2018.00474.

[16] M Shoaib et al. "An advanced deep learning models-based plant disease detection: A review of recent research". *Frontiers in Plant Science* 14 (2023), pp. 1158933–1158933.

[17] J Zhang, H Xia, and J G Yang. "Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art". *IEEE Geoscience and Remote Sensing Magazine* 9(2) (2021), pp. 8–32