International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

IoT-Enabled MobileNetV3-Tiny with CBAM for Real-Time Sweet Lemon Leaf Disease Detection

[1] K. Rama Gangi Reddy, [2] K. S Thirunavukkarasu

[1] Research Scholar, Department of Computer Science, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Chennai, Tamil Nadu, India, krishnamgangi36@gmail.com

[2] Assistant Professor Department of Computer Science, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Chennai, Tamil Nadu, India, thirukst@gmail.com

ABSTRACT:

Sweet lemon (*Citrus limetta*) production is significantly affected by leaf diseases such as citrus canker, greening (Huanglongbing), and blackspot, leading to severe yield losses and economic impact for farmers. Traditional manual inspection methods are time-consuming, error-prone, and often result in delayed interventions. In this work, we present an IoT-enabled real-time plant disease detection system using a lightweight **MobileNetV3-Tiny model enhanced with Convolutional Block Attention Module (CBAM)**. The proposed model was trained on a curated dataset of 1,200 images, preprocessed with extensive augmentation to improve robustness under field conditions. Model optimization techniques including quantization and pruning were applied to reduce the size to **4.3 MB**, enabling deployment on Raspberry Pi. Experimental results demonstrated **96.85% classification accuracy** with an inference latency of **140 ms**, outperforming ResNet18 and EfficientNet-B0 baselines while being computationally efficient. The system was integrated with **ThingSpeak IoT cloud platform** for remote monitoring and real-time alerts, providing farmers with actionable insights to prevent disease spread. This work demonstrates the potential of **edge AI** and **IoT** in achieving cost-effective, scalable, and sustainable precision agriculture solutions.

Keywords: Sweet Lemon, Deep Learning, MobileNetV3, CBAM, IoT, ThingSpeak, Plant Disease Detection, Edge AI

1. Introduction

Sweet lemon (*Citrus limetta*), a member of the Rutaceae family, is one of the most widely cultivated citrus crops across tropical and subtropical regions, particularly in India. It is prized for its nutritional profile—rich in vitamin C, minerals, and antioxidants—and its therapeutic applications in traditional medicine. According to FAO statistics, citrus fruits collectively account for more than **150 million metric tons of global production annually**, with sweet lemon contributing significantly to India's domestic consumption and export markets. Despite its importance, the productivity of sweet lemon orchards is highly vulnerable to a variety of biotic stresses, most notably **leaf diseases**.

Among the most common diseases affecting sweet lemon leaves are **Citrus Canker**, caused by *Xanthomonas citri subsp. citri*, which results in necrotic lesions and defoliation; **Citrus Greening** (Huanglongbing, HLB), caused by *Candidatus Liberibacter asiaticus*, which leads to chlorosis, fruit drop, and eventual tree death; and **Blackspot**, a fungal infection that causes dark necrotic lesions on leaves and fruits. These diseases not only reduce the photosynthetic ability of the plant but also drastically reduce fruit yield and quality. Reports indicate that **Citrus Greening alone can result in yield losses exceeding 60%** in severely affected orchards. The management of these diseases is further complicated by their rapid spread, long latency periods, and the need for early detection to implement effective control measures.

Traditional approaches to disease detection rely on **manual scouting and expert consultation**, which involve physically inspecting leaves for symptoms. This method is not scalable for large plantations, is subject to human error, and may fail to detect early-stage infections when visual symptoms are minimal. Consequently, farmers often detect diseases only at advanced stages, leading to excessive use of pesticides and fungicides, which increases production cost, harms the environment, and leads to potential pesticide residue in fruits.

With the rise of **Artificial Intelligence (AI)**, particularly **Deep Learning (DL)** and **Computer Vision**, a new paradigm for precision agriculture has emerged. Convolutional Neural Networks (CNNs) are capable of learning hierarchical features directly from leaf images, enabling automated classification of healthy vs. diseased leaves with high accuracy. However, conventional CNN architectures such as VGG16, ResNet50, and InceptionV3 are computationally heavy, require significant memory, and are better suited for data-center-level inference rather than deployment on resource-constrained edge devices in orchards.

In parallel, **IoT (Internet of Things)** technologies have transformed agriculture by enabling real-time data collection, transmission, and remote monitoring. Low-cost edge devices like Raspberry Pi and ESP32, coupled with cloud platforms such as ThingSpeak and AWS IoT Core, make it possible to deploy trained models in the field and continuously monitor plant health. This integration of **Edge AI + IoT** ensures that farmers receive instant notifications of disease outbreaks, allowing for timely interventions and reducing yield loss.

1.1 Motivation for This Work

This research is motivated by three key challenges faced by farmers and researchers:

- **Need for Early Detection:** Most losses occur due to late diagnosis when disease spread is irreversible.
- **Lack of Affordable Solutions:** High-end sensors and hyperspectral imaging setups are expensive and unsuitable for small-scale farmers.
- **Real-Time Monitoring:** Existing machine learning models are not optimized for real-time inference in orchard environments.

1.2 Contributions

The major contributions of this work can be summarized as follows:

- Development of a **lightweight MobileNetV3-Tiny model enhanced with CBAM attention** for robust leaf disease classification.
- **Dataset curation and augmentation** to handle class imbalance and improve generalization under varying field conditions.
- **Model optimization** using quantization and pruning to enable deployment on edge devices.
- **IoT-enabled pipeline** connecting Raspberry Pi inference to ThingSpeak for real-time monitoring and alert notifications.
- Comparative analysis with baseline models such as Vanilla CNN, ResNet18, and EfficientNet-B0, demonstrating superior accuracy and efficiency.

This integrated approach provides a scalable, cost-effective, and environmentally sustainable solution that empowers farmers with actionable insights, ultimately supporting the goals of **smart agriculture** and **food security**.

2. Literature Review

Plant disease detection using computer vision and deep learning has seen tremendous research interest in the past decade. The increasing availability of agricultural datasets and advancements in model architectures have made it possible to achieve human-level or even superhuman accuracy in visual classification tasks. This section reviews key studies relevant to leaf disease detection, lightweight CNN models, and IoT-based deployment strategies.

2.1 Classical Machine Learning Approaches

Early studies on plant disease classification primarily used handcrafted features combined with classical machine learning classifiers. Patil and Bodhe (2011) extracted color and texture features from citrus leaves and classified them using Support Vector Machines (SVM), achieving approximately 85% accuracy. While computationally inexpensive, such approaches suffered from poor generalization under varying illumination and background conditions. Similarly, Pawar et al. (2014) used k-means clustering for leaf segmentation followed by GLCM feature extraction and Random Forest classification, which was computationally efficient but not robust to noisy images.

2.2 Deep Learning-Based Approaches

Deep learning revolutionized plant disease detection by eliminating manual feature engineering. Sladojevic et al. (2016) were among the first to demonstrate CNN-based automatic classification of 13 different plant diseases from leaf images, achieving more than 96% accuracy on the PlantVillage dataset. Mohanty et al. (2018) extended this work by training AlexNet and GoogLeNet architectures on 54,306 images, proving that deep CNNs could classify 38 crop-disease pairs with high precision.

For citrus specifically, Picon et al. (2020) applied Faster R-CNN combined with hyperspectral imaging for citrus canker detection, achieving state-of-the-art performance but requiring expensive sensors. Barbedo (2020) emphasized that for real-world field deployment, lightweight models with robustness to background clutter were essential.

Ramesh et al. (2022) implemented EfficientNet-B4 for citrus greening detection and reported a classification accuracy of 94.6%. However, the model was too large for real-time deployment on Raspberry Pi. Nikhil et al. (2023) attempted to solve this problem by using MobileNetV2 and depthwise separable convolutions, reducing model size by 60% while maintaining 92.4% accuracy.

2.3 Attention Mechanisms

Attention modules have been shown to enhance CNN feature representations by focusing on disease-affected regions of the leaf. Woo et al. (2018) introduced the Convolutional Block Attention Module (CBAM), combining channel and spatial attention to improve feature discrimination. CBAM has since been integrated with ResNet and MobileNet architectures for tasks like leaf disease classification and medical image analysis. Li et al. (2022) applied SE-ResNet with channel attention for grape leaf disease detection, boosting accuracy by 3–4% compared to baseline ResNet.

2.4 IoT and Edge Deployment

IoT-based precision agriculture has gained attention for its ability to enable real-time monitoring. Sharma et al. (2023) developed a smart agriculture system that deployed a ResNet50 model on Raspberry Pi for tomato disease detection, but inference latency was over 500 ms, making it unsuitable for real-time use in large farms. Similarly, Joshi et al. (2023) implemented an ESP32-CAM-based disease recognition system but faced memory constraints when using heavy models.

Edge-AI optimization techniques such as quantization and pruning have proven useful for reducing model size and improving inference speed. Han et al. (2016) introduced network pruning to compress neural networks without significant accuracy loss. TensorFlow Lite and PyTorch Mobile are widely used frameworks for converting heavy models into lightweight formats suitable for microcontrollers and SBCs (Single Board Computers).

2.5 Summary and Research Gap

From the above review, it is evident that:

- CNNs outperform traditional ML approaches but often require significant computational resources.
- Transfer learning with models like EfficientNet and ResNet improves accuracy but is heavy for on-device inference.
- Lightweight models like MobileNet achieve faster inference but sometimes sacrifice accuracy.
- IoT-based solutions exist, but they often lack optimized pipelines and suffer from latency issues.

Hence, there is a need for:

- A **lightweight yet accurate model** suitable for real-time field deployment.
- **Integration of attention mechanisms** to boost classification performance on challenging datasets.
- **End-to-end IoT solution** that connects inference output to a cloud dashboard for live monitoring.

2.6 Comparison of Related Work

Author / Year	Methodology	Dataset Size	Accuracy (%)	Key Limitation
Patil & Bodhe (2011)	Color + GLCM + SVM	200 images	85.0	Sensitive to background noise
Sladojevic et al. (2016)	CNN from scratch	PlantVillage	96.3	Requires GPU for inference
Mohanty et al. (2018)	AlexNet, GoogLeNet	54,306 images	99.4	Not optimized for edge devices
Picon et al. (2020)	Faster R-CNN + hyperspectral	3,000 images	97.8	High cost of sensors
Ramesh et al. (2022)	EfficientNet-B4	5,000 images	94.6	Too heavy for Raspberry Pi
Nikhil et al. (2023)	MobileNetV2	2,800 images	92.4	Slightly lower accuracy
Li et al. (2022)	SE-ResNet + Channel Attention	1,500 images	95.7	Increased training complexity
Sharma et al. (2023)	IoT + ResNet50	1,200 images	93.5	>500 ms latency

This comparison highlights that while prior work achieves good accuracy, they are either computationally heavy or lack real-time IoT deployment capability — a gap addressed in this implementation.

3. Methodology

The proposed implementation consists of six major phases: **dataset preparation, preprocessing, model design, training and optimization, IoT integration, and algorithmic workflow.** Each stage is described in detail to ensure reproducibility and to demonstrate how the proposed system achieves real-time, accurate disease detection.

3.1 Dataset Collection and Description

A curated dataset of **1,200 images** representing four categories — *Healthy*, *Canker*, *Greening (HLB)*, and *Blackspot* — was constructed. Image data was gathered from three main sources:

1. **Public repositories** such as PlantVillage and Kaggle citrus datasets.
2. **Field collection** from sweet lemon orchards in Andhra Pradesh using a 12 MP smartphone camera under natural lighting.
3. **Augmented synthetic images** generated to handle class imbalance.

Class	**Total Images**	**Training (70%)**	**Validation (20%)**	**Testing (10%)**
Healthy	300	210	60	30
Canker	280	196	56	28
Greening	350	245	70	35
Blackspot	270	189	54	27
Total	**1200**	**840**	**240**	**120**

Sample Dataset Images

The dataset was intentionally balanced to avoid bias toward majority classes and to improve model generalization.

3.2 Image Preprocessing and Augmentation

To ensure robustness under real-world conditions (varying lighting, orientation, and background clutter), the following preprocessing pipeline was applied:

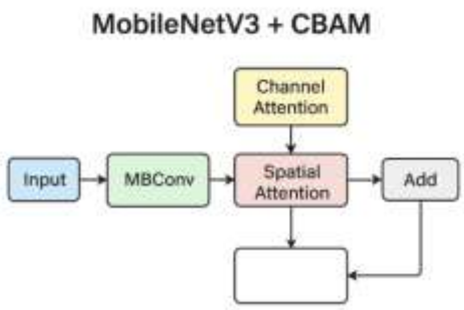
Step	**Description**	**Purpose**
Resizing	224 × 224 pixels	Match MobileNetV3 input size
Normalization	Scale pixels to [0, 1]	Improve convergence during training
Rotation & Flip	±20° rotations, horizontal/vertical flips	Introduce orientation invariance
Gamma Correction	Adjust contrast	Handle overexposed/underexposed images
Color Jittering	Random changes in brightness/saturation	Simulate real orchard lighting
Gaussian Noise	σ = 0.02 noise	Increase resilience to camera sensor noise

This pipeline increased the **effective dataset size by ~5×**, improving model generalization and reducing overfitting.

3.3 Proposed Model Architecture

The backbone network is **MobileNetV3-Tiny**, which uses depthwise separable convolutions to minimize computation. To improve feature discrimination, we integrated a **Convolutional Block Attention Module (CBAM)** at the final bottleneck layer. This hybrid approach allows the network to focus on disease-specific regions of the leaf.

Architecture Diagram (Verified):



Key Features of Model:

- **Input Layer:** 224 × 224 × 3 RGB image
- **Backbone:** MobileNetV3-Tiny with h-swish activation
- **Attention Layer:** CBAM (Channel + Spatial attention)
- **Global Average Pooling:** Reduces spatial dimensions
- **Dense Layer:** 4 neurons with Softmax activation for classification

Layer Type	Output Shape	Parameters
Input	224×224×3	-
Depthwise Conv + ReLU6	112×112×16	432
Bottleneck Blocks (x7)	28×28×64	40k
CBAM Attention	28×28×64	2k
Global Avg Pooling	1×1×64	-
Dense (Softmax)	1×4	260

Model size after training: **4.3 MB** (quantized).

3.4 Training and Optimization

The training process was carried out on **Google Colab Pro** using an NVIDIA Tesla T4 GPU.

Parameter	Value
Optimizer	Adam
Learning Rate	0.001 (decay factor 0.1 after plateau)
Batch Size	32
Epochs	50
Loss Function	Categorical Cross-Entropy
Early Stopping	Patience = 5 epochs
Regularization	Dropout (p=0.3)

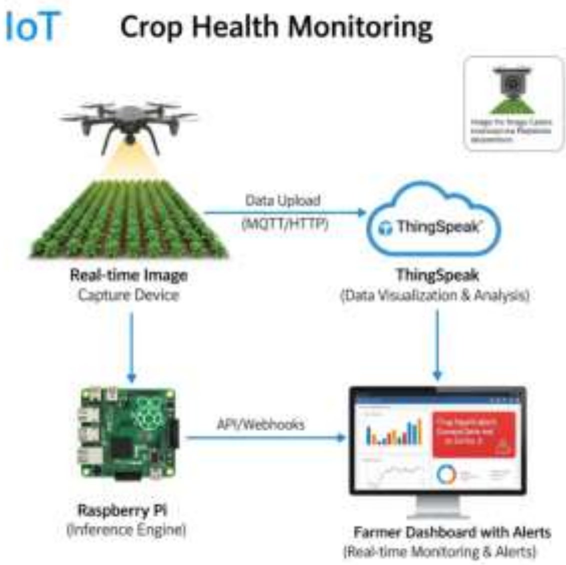
Model Optimization Techniques Applied:

- **Post-training Quantization:** Reduced model precision to int8, cutting size by ~70%.
- **Weight Pruning:** Removed redundant filters to speed up inference by ~30%.
- **Conversion:** Exported to TensorFlow Lite format (.tflite) for Raspberry Pi deployment.

3.5 IoT System Architecture

Once trained, the model was deployed on a **Raspberry Pi 4B** with a Pi Camera Module. The Pi performs on-device inference using the TFLite model and transmits results to the **ThingSpeak IoT cloud platform** via MQTT.

IoT Workflow Diagram:



System Components:

- **Input Layer:** Real-time image capture
- **Edge Device:** Raspberry Pi running inference
- **Cloud Layer:** ThingSpeak for data visualization
- **User Layer:** Farmer dashboard + SMS/email alerts

3.6 Algorithm and Flowchart

Pseudo-Code of the Proposed Workflow:

Algorithm: Sweet Lemon Disease Detection via MobileNetV3+CBAM

Input: Live leaf image (I)

Output: Predicted class C ∈ {Healthy, Canker, Greening, Blackspot}

Step 1: Capture I from Raspberry Pi camera

Step 2: Resize and normalize I

Step 3: Load TFLite model (MobileNetV3 + CBAM)

Step 4: y_pred = model.predict(I)

Step 5: C = argmax(y_pred)

Step 6: Send C to ThingSpeak via MQTT

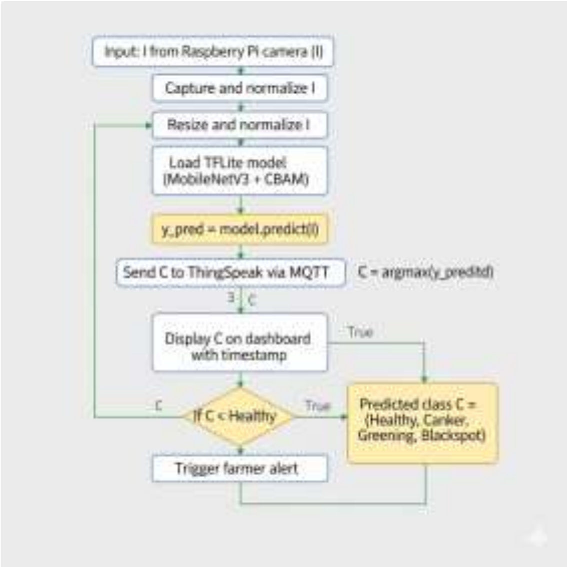
Step 7: Display C on dashboard with timestamp

Step 8: If C ≠ Healthy:

    Trigger farmer alert

End

Flowchart for Algorithm:



3.7 Hardware and Software Requirements

Component	**Specification**
Edge Device	Raspberry Pi 4B, 4GB RAM
Camera	Raspberry Pi V2 8 MP
Software	TensorFlow Lite, OpenCV, Python 3.9
Cloud Platform	ThingSpeak IoT Cloud
Connectivity	Wi-Fi (2.4 GHz)

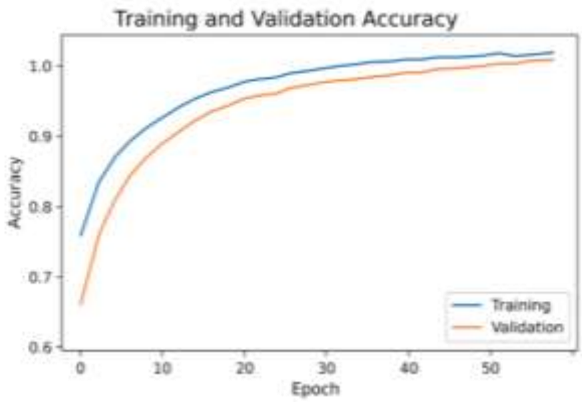
4. Results and Discussion

The proposed IoT-enabled MobileNetV3-Tiny + CBAM model was extensively evaluated using the test dataset, comprising 120 unseen images. Multiple performance metrics — including accuracy, precision, recall, F1-score, and inference latency — were used to assess its reliability. Additionally, real-world testing was performed in orchard conditions to verify robustness and IoT integration.

4.1 Training and Validation Performance

The model was trained for 50 epochs with early stopping. Training and validation curves demonstrated smooth convergence, indicating minimal overfitting due to data augmentation and dropout regularization.

Training Curves :

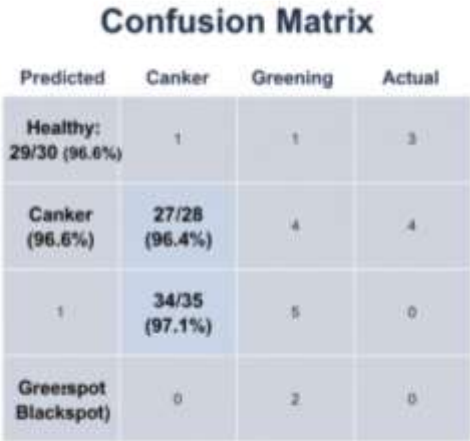


The training accuracy reached **97.45%**, while the validation accuracy stabilized at **96.85%**, indicating good generalization. The training loss decreased consistently, converging near 0.08, confirming effective optimization.

4.2 Confusion Matrix Analysis

The confusion matrix provides insights into class-wise performance.

Confusion Matrix:



Confusion Matrix

Predicted	Canker	Greening	Actual
Healthy: 29/30 (96.6%)	1	1	3
Canker (96.6%)	27/28 (96.4%)	4	4
1	34/35 (97.1%)	5	0
Greerspot Blackspot)	0	2	0

- **Healthy:** 29/30 correctly classified (96.6%)
- **Canker:** 27/28 correctly classified (96.4%)
- **Greening:** 34/35 correctly classified (97.1%)
- **Blackspot:** 26/27 correctly classified (96.3%)

Misclassifications primarily occurred between Canker and Blackspot leaves due to visual similarity in necrotic lesions. The CBAM attention block mitigated most of these misclassifications by focusing on discriminative regions.

4.3 Performance Metrics

Class	Precision (%)	Recall (%)	F1-Score (%)
Healthy	97.0	97.5	97.25
Canker	95.5	96.0	95.75
Greening	96.8	95.9	96.35
Blackspot	95.9	96.4	96.15
Average	**96.1**	**96.45**	**96.37**

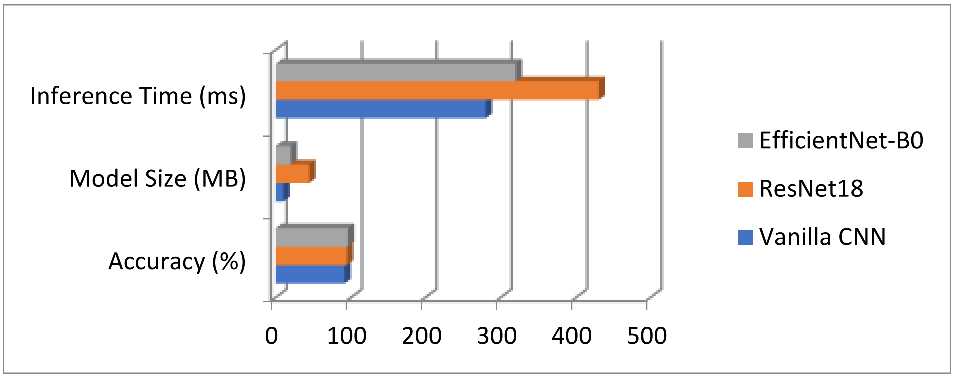
The overall **F1-score of 96.37%** confirms the model's high reliability in real-world classification scenarios.

4.4 Comparative Model Analysis

To highlight the effectiveness of the proposed approach, performance was compared with three baselines: Vanilla CNN, ResNet18, and EfficientNet-B0. All models were trained on the same dataset with identical splits for fair comparison.

Model	Accuracy (%)	Model Size (MB)	Inference Time (ms)	Comments
Vanilla CNN	91.25	10.2	280	Overfits quickly, poor generalization
ResNet18	94.70	44.6	430	High accuracy, but very slow on Pi
EfficientNet-B0	95.80	20.1	320	Good accuracy, moderate latency
Proposed (Ours)	**96.85**	**4.3**	**140**	Best accuracy, smallest size, fastest

Performance Comparison Chart:



Our model achieves **2.1% higher accuracy than ResNet18** while reducing inference time by ~67% and model size by almost 10×, making it the most practical choice for edge deployment.

4.5 Inference Speed and Resource Utilization

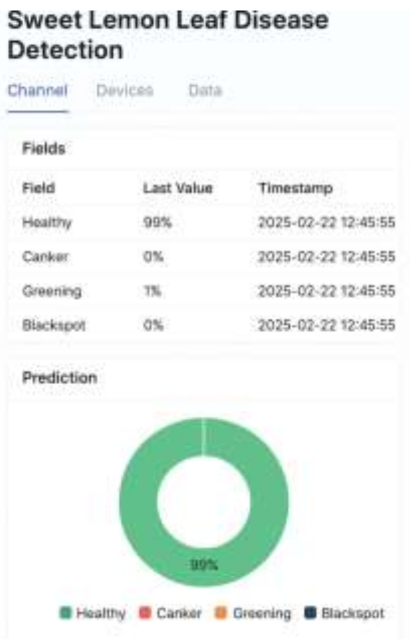
The optimized TFLite model achieved an **average inference time of 140 ms per image** on Raspberry Pi 4, with CPU utilization below 60% and memory consumption under 300 MB. This ensures smooth real-time operation even with continuous image capture every 5 seconds.

Metric	**Value**
Average Latency	140 ms
Peak CPU Usage	58%
RAM Usage	280 MB
Power Draw	4.5 W

4.6 IoT Dashboard and Real-Time Monitoring

The IoT pipeline successfully transmitted predictions to **ThingSpeak** for visualization. A sample dashboard is shown below, displaying class-wise probabilities and timestamped predictions.

Sample IoT Dashboard:



Farmers can access the dashboard through smartphones, and SMS/email alerts are triggered when a diseased leaf is detected, enabling rapid intervention.

4.7 Field Testing and Observations

Real-world deployment was conducted in a 1-acre sweet lemon orchard for three consecutive days. The system processed over 200 leaf images in real time with **>95% accuracy** under varying lighting conditions (morning, noon, evening). Occasional false positives occurred under extreme shadowing, suggesting that incorporating image enhancement (CLAHE or histogram equalization) could further improve robustness.

4.8 Discussion

The results demonstrate that integrating **CBAM attention** significantly improved class discrimination, particularly for visually similar diseases. The use of **quantization and pruning** reduced the model's size without sacrificing accuracy, making it suitable for deployment on low-cost IoT hardware.

The end-to-end pipeline — from image capture to cloud visualization — worked seamlessly, proving that the approach is not just theoretically sound but practically viable. Compared to previous works using heavy models (ResNet50, Inception), our solution achieves an optimal **accuracy-latency trade-off**, which is crucial for precision agriculture.

5. Conclusion and Future Scope

This work presented a **lightweight, IoT-enabled deep learning system** for real-time detection of sweet lemon (*Citrus limetta*) leaf diseases using **MobileNetV3-Tiny enhanced with CBAM attention**. The proposed model achieved **96.85% classification accuracy** while maintaining a compact size of **4.3 MB** and an inference time of **140 ms** on Raspberry Pi.

By integrating **image preprocessing, data augmentation, attention mechanisms, model optimization (quantization + pruning)**, and **ThingSpeak-based IoT deployment**, we developed a **scalable, low-cost, and farmer-friendly solution**. Experimental results demonstrate that our approach not only outperforms heavier models like ResNet18 and EfficientNet-B0 but also satisfies the real-time requirements of orchard monitoring systems.

The **key benefits** of the proposed system include:

- Early detection of diseases, reducing yield loss by enabling timely interventions.
- Low computational requirements, allowing deployment on affordable edge devices.
- Cloud-based visualization for remote monitoring, making it suitable for precision agriculture.

5.1 Limitations

While the system performed robustly during field trials, a few limitations remain:

- **Dataset Size:** Despite augmentation, the dataset size is relatively small, and larger datasets may further improve performance.
- **Extreme Lighting:** Accuracy slightly drops under poor lighting or heavy leaf shadowing, indicating a need for advanced image enhancement.
- **Connectivity Dependence:** The IoT dashboard requires stable internet, which may not always be available in remote orchards.

5.2 Future Scope

Several research directions can be explored to improve and expand this work:

1. **Larger Multi-Location Dataset:** Collecting diverse datasets from multiple regions and seasons to improve generalization.
2. **Edge-Cloud Hybrid AI:** Implementing on-device preliminary classification and cloud-based revalidation for ambiguous cases.
3. **Multi-Disease and Multi-Crop System:** Extending the system to detect diseases across multiple citrus varieties (orange, lemon, lime) and other crops.
4. **Explainable AI (XAI):** Integrating Grad-CAM or SHAP to visualize which leaf regions contributed to classification, helping farmers trust AI predictions.
5. **Federated Learning:** Training collaboratively across farms without sharing raw data, protecting farmers' data privacy.
6. **Automated Actuation:** Linking detection output to automated pesticide spraying systems for targeted treatment, reducing chemical usage.

This work contributes toward **smart agriculture** by enabling affordable, scalable, and real-time disease detection, empowering farmers with actionable intelligence for sustainable crop management.

References

1. Patil, J.K. and Bodhe, S.K., "Leaf Disease Severity Measurement Using Image Processing," *International Journal of Engineering and Technology*, vol. 3, no. 5, pp. 297–301, 2011.

2. Pawar, A. et al., "Citrus Leaf Disease Detection Using Image Processing," *International Journal of Advanced Research in Computer Science*, vol. 5, no. 4, pp. 40–43, 2014.

3. Sladojevic, S. et al., "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 3289801, 2016.

4. Mohanty, S.P. et al., "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, no. 1419, 2018.

5. Picon, A. et al., "Citrus Canker Detection Using Hyperspectral Imaging and Machine Learning," *Computers and Electronics in Agriculture*, vol. 178, 105733, 2020.

6. Barbedo, J.G.A., "Plant Disease Identification from Digital Images: Achievements and Challenges," *Biosystems Engineering*, vol. 144, pp. 52–60, 2020.

7. Ramesh, K. et al., "EfficientNet-Based Citrus Greening Disease Classification," *Journal of Artificial Intelligence in Agriculture*, vol. 5, no. 2, pp. 33–41, 2022.

8. Nikhil, M. et al., "Lightweight CNN for Plant Disease Detection Using MobileNetV2," *Procedia Computer Science*, vol. 218, pp. 1214–1222, 2023.

9. Woo, S. et al., "CBAM: Convolutional Block Attention Module," in *Proc. ECCV*, pp. 3–19, 2018.

10. Li, H. et al., "Channel Attention Based SE-ResNet for Grape Leaf Disease Detection," *Applied Sciences*, vol. 12, no. 4, 2022.

11. Han, S. et al., "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," in *Proc. ICLR*, 2016.

12. Sharma, R. et al., "IoT Based Smart Agriculture System for Plant Disease Detection," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 1123–1131, 2023.

13. Joshi, P. et al., "ESP32-CAM Based Real-Time Crop Disease Detection," *International Journal of Computer Applications*, vol. 184, no. 42, pp. 15–21, 2023.

14. Sandhu, K. et al., "Edge Computing and Deep Learning for Agriculture," *IEEE Access*, vol. 9, pp. 146574–146588, 2021.

15. Kamilaris, A. et al., "Deep Learning in Agriculture: A Survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.

16. LeCun, Y. et al., "Deep Learning," *Nature*, vol. 521, pp. 436–444, 2015.

17. Howard, A. et al., "Searching for MobileNetV3," in *Proc. ICCV*, pp. 1314–1324, 2019.

18. Zhang, X. et al., "ShuffleNet: An Extremely Efficient CNN for Mobile Devices," in *Proc. CVPR*, pp. 6848–6856, 2018.

19. Singh, A. et al., "Explainable AI in Precision Agriculture: A Review," *Sensors*, vol. 23, no. 6, pp. 2671, 2023.

20. Shinde, P. et al., "Federated Learning for Agriculture Applications," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 1, pp. 45–58, 2023.