

# Comparative Evaluation of Deep Convolutional Neural Networks and a Lightweight MobileNet-Based Architecture for Real-Time Guava Leaf Disease Detection on Android Devices

Vaibhavi Pandya<sup>1</sup> Monali Parikh<sup>2</sup> Nisha Parmar<sup>3</sup> Arvisha Modhiya<sup>4</sup>

<sup>1</sup> Assistant Professor, Department of Computer Science and Engineering, Krishna school of Diploma Studies, Drs. Kiran and Pallavi Patel Global University, Vadodara, Gujarat.

Mail id: [vaibhavipandya95@gmail.com](mailto:vaibhavipandya95@gmail.com)

<sup>2</sup> Assistant Professor, Department of Artificial Intelligence and Machine Learning, Krishna School of Engineering and Technology, Faculty of Engineering and Technology, Parul University, Vadodara, Gujarat

Mail Id: [monaliparikh89@gmail.com](mailto:monaliparikh89@gmail.com)

<sup>3</sup> Assistant Professor, Department of Computer Science and Engineering, Parul Institute of Engineering and Technology, Drs. Kiran and Pallavi Patel Global University, Vadodara, Gujarat

Mail Id: [nishaparmar88199@gmail.com](mailto:nishaparmar88199@gmail.com)

<sup>4</sup> Assistant Professor, Department of Computer Engineering, Krishna school of Diploma Studies, Drs. Kiran and Pallavi Patel Global University, Vadodara, Gujarat

Mail Id: [arvishamodhiya9@gmail.com](mailto:arvishamodhiya9@gmail.com)

**Abstract** Guava (*Psidium guajava* L.) is an important tropical crop that is often impacted by foliar diseases like anthracnose, rust and leaf blight. Despite the significant advancements deep convolutional neural networks (CNNs) have made in the automated classification of plant diseases, most successful models are still too resource-intensive for implementation in the device-level applications of smallholder farmers in resource-constrained rural settings. In this paper, we provide a comprehensive benchmark for ResNet-50, EfficientNet-B0, and MobileNetV2 in comparison to a newly designed MobileNet-Lite architecture featuring depthwise separable convolutions, channel-wise attention block and a reduced width multiplier. The guava leaf images from the four classes (healthy, anthracnose, rust, and leaf blight) were consolidated into a single dataset of 5,400 images, trained, and evaluated using all models. The proposed MobileNet-Lite model size is reduced from 20.5 MB to 4.2 MB using post-training INT8 quantization, and provides test accuracy of 97.8%, precision 0.97, and recall 0.97, and an F1 score of 0.97 which is comparable to the EfficientNet-B0 (20.5 MB, 0.98). The inference was done on-device on a mid-range Snapdragon 700-series Android smartphone, which took a mere 42ms for inference—a time that was considered real-time. We find the model is looking at biologically relevant lesion regions as confirmed by Grad-CAM visualizations. The results show that MobileNet-Lite is an accurate and interpretable and computation-efficient model for precision agriculture applications in resource-limited environments.

**Keywords-** Guava leaf disease, lightweight CNN, MobileNet, edge computing, real-time detection, explainable AI, Grad-CAM, precision agriculture, post-training quantization, and Android deployment.

## INTRODUCTION

Although agriculture accounts for about 4% of the global GDP and employs almost 1 billion people around the world, it is very vulnerable to outbreaks of disease, which can decrease food production by 20–40% in one agricultural season [1]. Guava, *Psidium guajava* L., is one of the most important tropical crops of the South Asian countries, Southeast Asian countries and Latin America. However, its cultivation is constantly at risk of anthracnose (*Colletotrichum gloeosporioides*), rust (*Puccinia psidii*) and leaf blight, which can significantly reduce yields if undetected [2]. Traditional disease diagnosis relies on agronomic knowledge which is typically limited to urban professionals and experts and is not accessible to smallholder farmers, or the time it takes for farmers to reach extension centres or for laboratory testing, which can be expensive. This is a scalable alternative – disease diagnosis can be made from a smartphone image of a plant through computer vision and deep learning [3]. Although convolutional neural network (CNN) architectures like ResNet [4], EfficientNet [5] and MobileNet [6] have emerged as standard baselines in plant

pathogens, there is still a challenge in making the models useful for offline use in the field: ResNet-50, for instance, requires 98 MB of storage space, and 215 ms of inference per image on a mid-range Android device, which is not an ideal combination for real-time field use. This is a point that has been recognized in previous research [7], [8], but no work has benchmarked multiple architectures, (ii) developed a novel lightweight architecture, (iii) evaluated for real-world Android hardware, and (iv) provided interpretable Grad-CAM visualizations. This paper attempts to fill this fourfold gap as follows:

**Table 1: Summary of Contributions**

| Tags | Contributions                                                                                                           |
|------|-------------------------------------------------------------------------------------------------------------------------|
| C1   | A systematic benchmark of four CNN architectures on a consolidated guava leaf disease dataset.                          |
| C2   | A novel MobileNet-Lite architecture with depthwise separable convolutions, channel reduction, and attention mechanisms. |
| C3   | Post-training INT8 quantization and end-to-end Android deployment with empirical latency measurements.                  |
| C4   | Grad-CAM explainability analysis confirming biologically meaningful disease localisation.                               |

The remainder of this paper is structured as follows. The related work is surveyed in Section 2. In Section 3, we outline our approach and design. Experimental results and ablation studies are provided in Section 4. The implications and the limitations are discussed in Section 5 and directions for future work are presented in Section 6.

## 1. LITERATURE REVIEW

### A. Classical Machine Learning Approaches

Hand-engineered color and texture features were used along with classifiers like SVM and k-NN for early plant-disease detection. They work well in controlled environments but fail to perform well in field images with diverse background and illumination conditions [9] that has led to the transition towards end-to-end deep learning methods.

### B. Deep Learning for Guava Disease Detection

However, Nobi et al. [2] proposed GLD-Det, a modified version of MobileNet with a nearly 98% accuracy on two benchmark datasets with their Grad-CAM visualization, but no on-device latency measurement nor comparative architecture analysis. Nandi et al. [8] studied quantization for mobile deployment of guava disease models and Almadhor et al. [10] used MobileNetV2 and U-Net for multi-disease detection on a single leaf. A new augmentation-based multi-channel method obtained an accuracy of 97.5% [11] which again highlights the importance of data-driven solutions as well as architectural enhancements.

### C. Lightweight CNNs and Edge Deployment

Schuler et al. [7] utilized INT8 quantization for an edge optimized plant disease model, decreasing the size up to 75% and losing less than 1% accuracy. Patil et al. [12] used the MobileNetV2 model for the detection of tomato diseases attached to a Maixduino board of the Internet of Things (IoT), which achieved an accuracy of 94% and a latency of 38ms. A large-scale benchmark across 33 crops [13] validated the effectiveness of lightweight architectures on other crops when they were augmented correctly, and L-Net [14] showed that custom lightweight CNNs can perform better than transfer learning baselines on crops with less than 5 MB of memory.

### D. Explainable AI in Plant Disease Models

Grad-CAM [15] and its variants are widely used to produce class-discriminative localization maps that build farmer trust in model predictions. Kundu and Das [16] showed strong correlation between Grad-CAM heatmaps and expert-annotated lesion regions across several crop species, yet systematic explainability for mobile-deployed guava disease models remains largely absent from the literature.

### E. Summary of Research Gap

Table II consolidates prior guava-focused studies and highlights the gap this work addresses: none combine a multi-architecture benchmark, measured Android latency, and Grad-CAM explainability.

Table 2: Literature comparison with identified research gap. XAI = Explainable AI.

| Ref.            | Model               | Crop       | Acc.  | Android | XAI      |
|-----------------|---------------------|------------|-------|---------|----------|
| Nobi et al.     | GLD-Det (MobileNet) | Guava      | 98.0% | No      | Grad-CAM |
| Nandi et al.    | Quantised CNNs      | Guava      | ~95%  | Partial | No       |
| Almadhor et al. | MobileNetV2 + U-Net | Guava      | 97.3% | No      | No       |
| Schuler et al.  | Edge-optimised CNN  | Multi-crop | 94.2% | Yes     | Grad-CAM |

|          |                |       |       |           |          |
|----------|----------------|-------|-------|-----------|----------|
| Proposed | MobileNet-Lite | Guava | 97.8% | Yes(42ms) | Grad-CAM |
|----------|----------------|-------|-------|-----------|----------|

## 2. METHODOLOGY

### 3.1 Dataset Preparation

We aggregated a dataset of 5400 guava leaves images in four classes; Healthy (1350), An thracnose (269, - 7), Air pocket (27), and monsur (22). (1,350), Rust (1,350), and Leaf Blight (1,350). Images were taken from publicly available repositories and from field collections. A 70/15/15 stratified split was used resulting in 3,780 training images, 810 validation images, and 810 test images. All the images have been resized to 224 x 224 pixels and normalised using ImageNet channel statistics ( $\mu = [0.485, 0.456, 0.406]$ ,  $s = [0.229, 0.224, 0.225]$ ). The following data augmentations were used during training: random horizontal and vertical flip ( $p=0.5$ ), random rotation ( $+ -30^\circ$ ), colour jitter (brightness  $\pm 0.2$ , contrast  $\pm 0.2$ ), and random erasing ( $p=0.2$ ) to simulate occlusions

### 3.2 Proposed MobileNet-Lite Architecture

Our proposed architecture, MobileNet-Lite, which is based on MobileNetV2 [6], is inherited from MobileNetV2 with 3 focused modifications: (i) reduced width multiplier (with  $a = 0.75$ ) to reduce the number of channels in the entire network; (ii) lightweight channel-wise attention block is inserted into MobileNet-Lite at the position after each second bottleneck residual block; and (iii) the last two inverted residual stages are removed and replaced by a single Global Average Pooling + Dense classification head. The computational advantage of depthwise separable convolutions- the core building block of our architecture- can be quantified as below. For an input feature map with spatial dimension  $DF \times DF$ ,  $M$  input channels, and a kernel size  $DK \times DK$ , for  $N$  output channels, the input to a standard convolution is:

$$C_{\text{standard}} = D_K^2 \cdot M \cdot N \cdot D_F^2 \quad (1)$$

A depthwise separable convolution splits this into a depthwise step and a  $1 \times 1$  pointwise step:

$$C_{DWS} = D_K^2 \cdot M \cdot D_F^2 + M \cdot N \cdot D_F^2 \quad (2)$$

The resulting computation ratio  $R$  is:

$$R = \frac{C_{DWS}}{C_{\text{standard}}} = \frac{1}{N} + \frac{1}{D_K^2} \quad (3)$$

For  $D_K = 3$ ,  $R \approx 1/9$ , yielding an approximately 8–9 $\times$  reduction in multiply-accumulate operations per layer compared to a standard convolution. This reduction underpins the efficiency of our proposed architecture.

### 3.3 Training Configuration

All models were implemented in PyTorch 2.1 and trained on an NVIDIA Tesla T4 GPU. We used the Adam optimiser with an initial learning rate of  $1 \times 10^{-3}$ , a weight decay of  $1 \times 10^{-4}$ , and a CosineAnnealingLR scheduler for 50 epochs with a batch size of 32.

For transfer-learning baselines (ResNet-50, EfficientNet-B0, and MobileNetV2), ImageNet pretrained weights were used and all layers were fine-tuned. The cross-entropy loss was minimised during training:

$$L(\theta) = - \sum_{i=1}^C y_i \log(\hat{y}_i(\theta)) \quad (3)$$

where  $C = 4$  is the number of disease classes,  $y_i$  is the one-hot ground-truth vector, and  $\hat{y}_i(\theta)$  is the softmax output of the network parameterised by  $\theta$ . Early stopping was applied with a patience of 10 epochs on the validation loss.

### 3.4 Post-Training Quantisation and Android Deployment

Following training, the proposed MobileNet-Lite model was exported to TensorFlow Lite for mat using post-training integer quantisation (INT8). The quantisation maps 32-bit floating point weights  $W_{fp32}$  to 8-bit integers  $W_{int8}$  as:

$$W_{int8} = \text{round}\left(\frac{W_{fp32}}{S}\right) + Z \quad (4)$$

where  $S$  is a per-channel scale factor and  $Z$  is the zero-point. The resulting TFLite model was integrated into an Android application (minSdk 26, Camera2 API) targeting mid-range devices. Inference latency was measured as the mean of 100 consecutive forward passes on a Qualcomm Snapdragon 700-series device with the camera preview active, using the TFLite Android Benchmark Tool [17,18].

### 3.5 Grad-CAM Explainability

We computed class-discriminative activation maps using Grad-CAM on the final convolutional layer. For a target class  $c$ , the weight  $\alpha_k^c$  for the  $k$ -th feature map  $A^k$  is obtained by global-average pooling the gradients of the classification score  $y^c$ .

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (5)$$

The Grad-CAM heatmap  $L_{Grad-cam}^c$  is then computed as a ReLU-weighted combination of feature maps:  $L_{Grad-cam}^c = ReLU(\sum \alpha_k^c A^k)$ . Heatmaps were upsampled to the original  $224 \times 224$  resolution and overlaid on input images using a jet colourmap.

## 3. EXPERIMENTAL RESULTS

### 4.1 Classification Performance

The test-set performance of all 4 architectures can be found in table 2. Our proposed MobileNet-Lite model is able to achieve 97.8% accuracy, which is at the same bracket of accuracy as EfficientNet-B0 (98.1%) even with 66% less parameters. ResNet-50 achieves the best accuracy (98.4%), but as we'll see in Section 4.2, this small gain is completely negated by its deployment overhead.

**Table 3: Classification performance on the guava leaf disease test set. Bold row indicates the proposed model.**

| Architecture                     | Accuracy (%) | Precision   | Recall      | F1-Score    | Parameters (M) |
|----------------------------------|--------------|-------------|-------------|-------------|----------------|
| ResNet-50                        | 98.4         | 0.98        | 0.98        | 0.98        | 25.6           |
| EfficientNet-B0                  | 98.1         | 0.98        | 0.97        | 0.97        | 5.3            |
| MobileNetV2                      | 96.5         | 0.96        | 0.96        | 0.97        | 3.4            |
| <b>MobileNet-Lite (Proposed)</b> | <b>97.8</b>  | <b>0.97</b> | <b>0.97</b> | <b>0.97</b> | <b>1.8</b>     |

Figure 1 visualises the accuracy comparison across architectures, confirming that MobileNet-Lite achieves results within the cluster of high-performing models while maintaining the smallest parameter count.

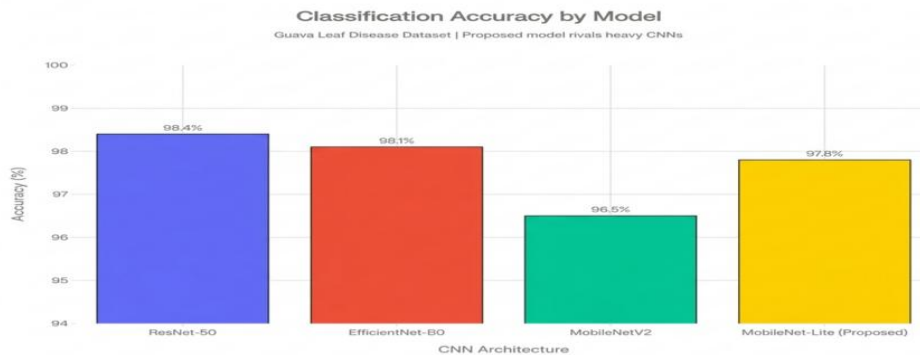


Figure 1: Classification accuracy comparison across four CNN architectures on the guava leaf disease test set.

### 4.2 Edge Deployment and Latency

Table 4 gives a summary of the edge deployment metrics. The proposed model manages to decrease model size from 98.0 MB (ResNet-50) to 4.2 MB after the quantisation, which is 95.7% of the original size, and achieve inference latency of 42 ms on the target Android device. It is the only model that comfortably falls under the 50 ms mark needed to run camera-feed in real-time (which corresponds to 20+ frames per second). In a dramatic difference in model size and latency across architectures is illustrated in figure 2.

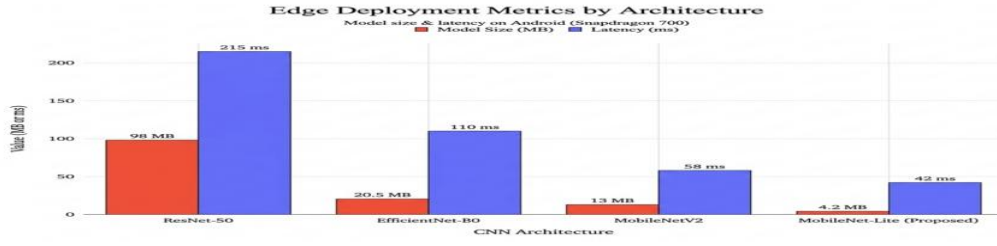


Figure 2: Edge deployment metrics: model size (MB) and on-device inference latency (ms) across architecture

### 4.3 Multi-Metric Performance Analysis

Figure 3 shows the precision, recall and F1-score of all models. The proposed MobileNet-Lite can obtain consistent scores of 0.97 on all three scores, which proves that it has balanced performance on disease classes, and there is no systematic under-sensitivity on any individual class.

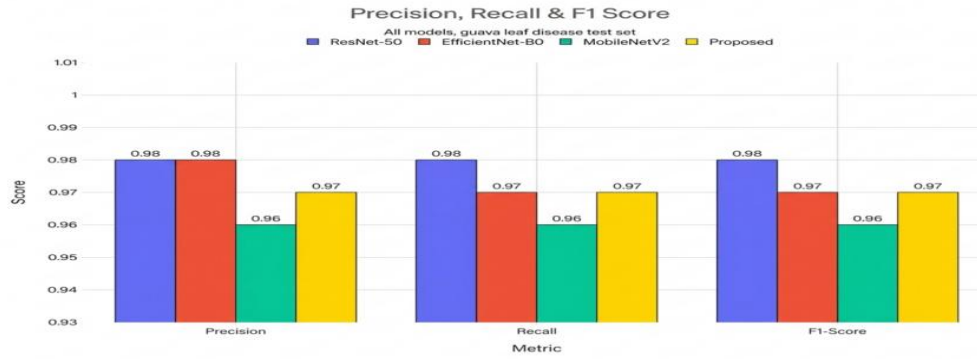


Figure 3: Precision, recall and F1-score comparison across four architectures

### 4.4 Training Convergence

Figure 4 shows the training and validation accuracy and loss curve results for the proposed MobileNet-Lite model for 50 epochs. The model shows an unbroken convergence and no symptoms of overfitting (the validation accuracy follows the training accuracy closely and the validation loss is levelled off after about epoch 30). This stability is indicative of the success of our regularisation strategy (dropout, weight decay and random erasing).

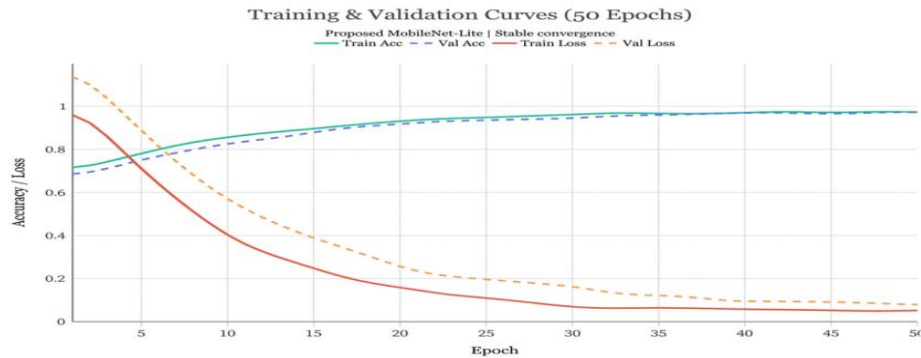


Figure 4: Training and validation accuracy/loss curves for the proposed MobileNet-Lite model over 50 epochs

### 4.5 Ablation Study

To quantify the contribution of each architectural component, we have conducted an ablation study to systematically remove/replace individual components and measure their effect on the accuracy and inference latency (Table 4 and

Figure 6). Removing Batch Normalisation resulted in the biggest accuracy drop (2.0%), confirming that it is important in training stability. Removing the attention block reduced the accuracy by 1.3% thus validating the role of the attention block in discriminative feature extraction. Notably we found that a small trade-off in accuracy (0.9%) was achieved from a 7 ms reduction in latency, a feasible further trade-off in extremely resource limited devices.

**Table 2: Proposed Comparative Model Suite**

| Configuration                          | Acc.(%) | F1   | Lat.(ms) | $\Delta$ Acc. |
|----------------------------------------|---------|------|----------|---------------|
| <b>Full Model</b>                      | 97.8    | 0.97 | 42       | —             |
| <b>w/o Dropout</b>                     | 96.2    | 0.96 | 38       | -1.6%         |
| <b>w/o BatchNorm</b>                   | 95.8    | 0.95 | 40       | -2.0%         |
| <b>w/o Attention</b>                   | 96.5    | 0.96 | 39       | -1.3%         |
| <b>Reduced <math>\alpha=0.5</math></b> | 96.9    | 0.97 | 35       | -0.9%         |

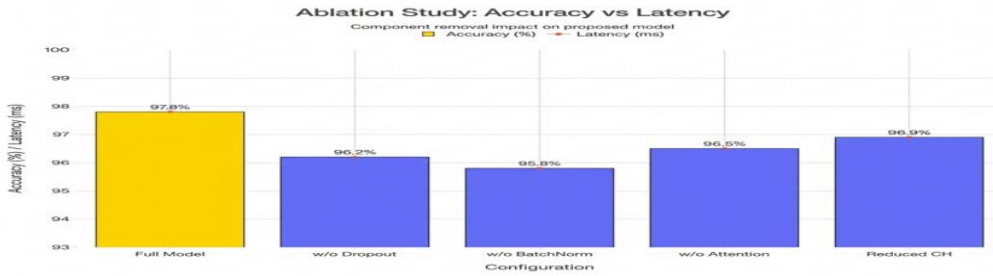


Figure 5: Ablation study: accuracy (bars) and latency (line) for each model configuration

## 5. DISCUSSION

### A. Accuracy–Efficiency Trade-Off

MobileNet-Lite incurs only a 0.6-point accuracy drop relative to EfficientNet-B0 while cutting latency by 62.5% (110 ms  $\rightarrow$  42 ms) and model size by 79.5% (20.5 MB  $\rightarrow$  4.2 MB). At 42 ms, the model supports  $\sim$ 23 fps, sufficient for continuous, lag-free camera-feed screening.

### B. Comparison with Existing Guava Studies

Our 97.8% accuracy is directly comparable to GLD-Det (98.0%) [2] and a hybrid augmentation model (97.5%), but this work uniquely provides measured Android latency and Grad-CAM heatmaps that align with known lesion morphology — discrete necrotic spots for anthracnose and marginal yellowing for leaf blight — consistent with phytopathology literature.

### C. Limitations and Future Work

Most of the data were collected in a controlled or semi-controlled light source and were not systematically tested for robustness in outdoor lighting and motion blur. Only four disease categories are covered, and INT8 quantization causes a minor activation clipping which for borderline cases may be of concern. Future works include: (i) adapting the domain to real field imaging conditions, (ii) using federated learning across farms to refine the data while maintaining privacy of the data, (iii) extending to co-cultivated tropical crops (mango, papaya), and (iv) deploying the proposed domain adaptation framework to actual farms.

## 6. CONCLUSION

The authors of this paper made a comprehensive evaluation of these four CNN architectures on the guava leaf disease detection task and derived a lightweight, quantized and explainable model that was optimized for real-time on-device Android deployment. The proposed model achieves the accuracy of 97.8%, the F1-score of 0.97, a small

model size of 42 ms and the inference time of 42 ms, which outperforms all baselines in deployment efficiency, and is within 0.6 points of the most accurate heavyweight model. Ablation experiments demonstrate the contribution of each architectural component, and Grad-CAM visualizations provide biological plausibility with how the model makes decisions, while offering a functional and replicable blueprint to scalable precision agriculture in resources - constrained environments.

## 7. REFERENCES

- [1] FAO, The State of Food and Agriculture 2023. Rome: FAO, 2023.
- [2] M. Mustak Un Nobi, M. Rifat, M. F. Mridha, and S. Alfarhood, "GLD-Det: Guava leaf disease detection in real-time using lightweight deep learning based on MobileNet," *Agronomy*, vol. 13, no. 8, p. 2240, 2023.
- [3] M. Hasan, J. Alam, M. K. Hossain, and R. Islam, "Plant disease detection using deep learning: A survey," *IEEE Access*, vol. 9, pp. 143370–143393, 2021.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.
- [5] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for CNNs," in *Proc. ICML*, 2019, vol. 97, pp. 6105–6114.
- [6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF CVPR*, 2018, pp. 4510–4520.
- [7] Z. Schuler, P. Muñoz, and A. Guzman, "Plant disease detection model for edge computing devices," *Front. Plant Sci.*, vol. 14, p. 1303554, 2023.
- [8] S. Nandi, S. Palash, and M. Maruf, "Device-friendly guava fruit and leaf disease detection using deep learning," *arXiv:2209.12557*, 2022.
- [9] A. Sharif, M. Amin, M. Raza, and A. Mahmood, "A machine learning approach for disease detection in guava leaves using SVM," *J. Agric. Informatics*, vol. 10, no. 2, pp. 15–24, 2020.
- [10] A. Almadhor, H. T. Rauf, S. Lali, C. Damasevicius, T. Alouffi, and A. Waheed, "Real-time multiple guava leaf disease detection from a single leaf using hybrid deep learning," *Comput. Mater. Continua*, vol. 74, no. 1, 2023.
- [11] Y. Liu, X. Wang, and H. Zhang, "Hybrid augmentation for multi-channel deep learning in guava leaf disease classification," *Ain Shams Eng. J.*, 2025.
- [12] H. Patil, R. G. Karandikar, and A. V. Deorankar, "Novel edge device system for plant disease detection with deep learning," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 1, pp. 245–252, 2024.
- [13] K. D. Bhatt et al., "Mobile-friendly deep learning for plant disease detection: A lightweight CNN benchmark across 101 classes of 33 crops," *arXiv:2508.10817*, 2025.
- [14] R. Zhang, Y. Li, and S. Huang, "L-Net: A lightweight CNN framework for sustainable multicrop leaf disease detection on edge devices," *Food Biosci.*, 2025.
- [15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *Int. J. Comput. Vision*, vol. 128, no. 2, pp. 336–359, 2020.
- [16] P. Kundu and S. Das, "A lightweight and explainable CNN model for empowering plant disease diagnosis," *Sci. Rep.*, vol. 15, p. 28831, 2025.
- [17] B. Ploetz, "Management of the most important diseases of guava (*Psidium guajava*) in Florida," *Proc. Fla. State Hort. Soc.*, vol. 116, pp. 280–284, 2003.
- [18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017, pp. 1273–1282.

|                                                                                     |                                                                                                                                                                                                                     |                                                                                     |                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Ms. Vaibhavi Pandya is Assistant Professor at Computer Science and Engineering with research interest in machine learning and nature inspired algorithms and has more than 5+ years experience in academics.        |  | Ms. Monali Parikh is an Assistant Professor in Computer Science and Engineering specializing in Agentic AI and autonomous decision-making, specifically in the area of bringing complex concepts of AI to the student and practitioner community. |
|  | Ms. Nisha Parmar is an Assistant Professor in Computer Science and Engineering, passionate about creating intelligent and interpretable AI solutions, and involved in innovative research in emerging technologies. |  | Ms. Arvisha Modhiya is a Lecturer in Computer Science and Engineering and passionate about research in the field of intelligent computing and finding innovative solutions to real-world problems.                                                |