Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal, 2026, 20(8), 476–486
https://doi.org/10.12913/22998624/220516
ISSN 2299-8624, License CC-BY 4.0

Received: 2026.01.10
Accepted: 2026.04.09
Published: 2026.07.01

Comparative analysis of various lightweight deep learning methods to diagnose the real-time chilli leaf diseases on edge devices

Vishali Sivalenka[1*], Rekha Gillala[1]

[1] Koneru Lakshmaiah Education Foundation, Hyderabad, Telangana, India
* Corresponding author's e-mail: vishali.sivalenka@klh.edu.in

ABSTRACT

Farmers deep learning models used in precision agriculture are usually limited by the few computational resources of edge devices. Although many convolutional neural network (CNN) architectures are available, it is not clear through rigorous benchmarking which model would provide the best trade-off between accuracy and latency for a specific crop like Chilli (*Capsicum annuum*) that has complicated pathological features like leaf curling. In this paper, the authors supported different benchmark tests that have been run with cutting-edge lightweight architectures: MobileNetV3, ShuffleNetV2, and EfficientNet-B0 – versus the heavy baseline ResNet-50 and a new attention-enhanced framework (Effi-AttnNet). In order to compare the performance of different models, training and testing of all of them were conducted under similar hyperparameters on a dataset comprising six classes of chilli diseases. The comparative study results highlight the large differences in the models' performance: For example, ShuffleNetV2 was able to produce an accuracy as high as 98.66% at a very fast inference speed (122 FPS), whereas the widely used MobileNetV3 was hardly able to generalize and therefore its accuracy was only 81.23%. Effi-AttnNet, which is authors custom-built framework, performs better compared to the above-mentioned models with an accuracy of 99.73%. Such a high precision result combined with the model's capability to be efficiently deployed at the edge and essentially achieves an inference speed of 60.3 FPS. This benchmarking work has been instrumental in providing essential information to the decision-makers in choosing the ideal architecture for on-the-go, mobile-based plant disease diagnosis systems.

Keywords: deep learning, edge computing, benchmarking, MobileNetV3, ShuffleNetV2, EfficientNet, chilli leaf disease.

INTRODUCTION

Chilli (*Capsicum annuum*) is one of the major commercial spice crops in India, which makes a significant contribution to the country's agricultural economy, mainly in the districts of Telangana and Andhra Pradesh. Unfortunately, the crop yield has been lowered due to biotic stresses including viral diseases, like Leaf Curl (Begomovirus), and fungal diseases, like Cercospora Leaf Spot [1]. Quick detection is what can help a lot in saving the crop, however, traditional manual scouting, albeit being a time-consuming and labor-intensive process, is also very subjective and often incorrect [2] (Figure 1).

Introducing artificial intelligence (AI) and deep learning (DL) technologies is a modern solution in line with the agriculture 4.0 era for automatically detecting diseases. It has been proven that the image classification tasks become easy by using convolutional neural networks (CNNs) as they reach the human-level accuracy [3]. The initial change of models in this sector had something to do with heavy architectures like VGG-16 and ResNet-50 [4]. Although very high accuracies could be achieved by such models, they may not be readily available because of the high computational power requirements needed to run them. A case in point is VGG-19, which has more than 140 million parameters

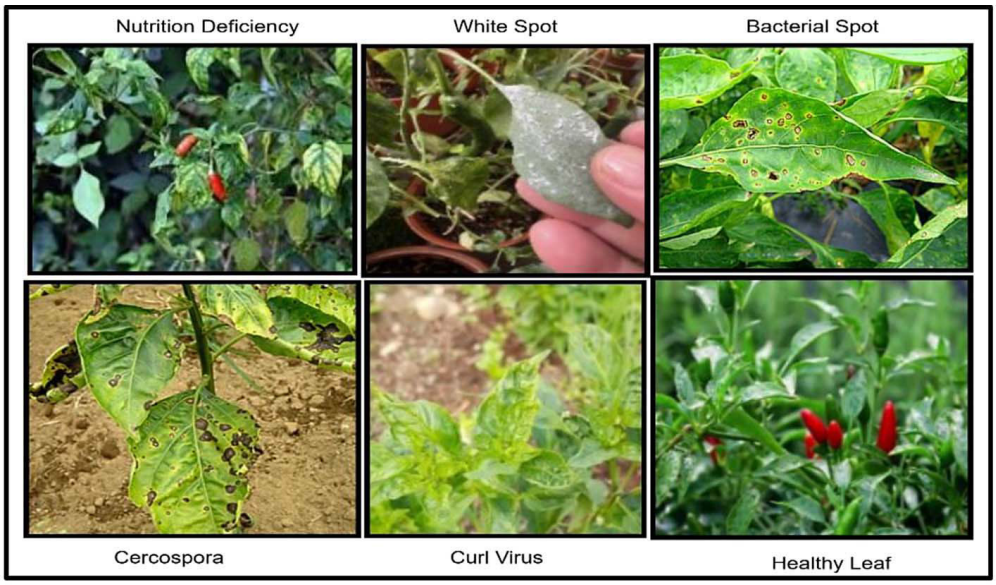

Figure 1. Chilli leaf diseases

making it almost impossible to run on devices with limited resources, such as smartphones or IoT-enabled drones [5].

Lightweight CNN models with smaller sizes were invented by the community of researchers as a solution to the problem of hardware limitations of edge devices to the issue of mobile inference most efficiently. MobileNet [6], ShuffleNet [7], and EfficientNet [8] are the most cited examples of such models. To a great extent, these models bring about reductions in model size and latency by means of Depthwise separable convolutions, channel shuffling, and compound scaling techniques, respectively. Zhang et al. [9] demonstrated the potential of MobileNetV3 for the detection of potato diseases, thus making the right trade-off between speed and accuracy. Accordingly, Rashid et al. [10] have employed the modified MobileNet architectures for tomatoes.

Still, there is a significant difference that has not been addressed in the comparison of these models in a Chilli crop context. The infected leaf with the help of the leaf curl virus of the potato or tomato leaves, normally being flat, the chilli leaf has complex 3D geometric deformations [11]. It is not clear whether typical lightweight models can handle the morphological complexities of this kind. Besides that, most of the existing works present their results in terms of accuracy, while they do not have a detailed discussion about the inference speed (FPS) on edge hardware, which is the main factor for the usability in the real world [12].

This study is a concerted effort to bridge the gap in comparing various deep learning architectures with authors' own homegrown system, Effi-AttnNet. In contrast with traditional attention-augmented networks, which add modules in every single layer, Effi-AttnNet can use a so-called Strategic Decoupling strategy. The attention control was confined at several phases: Channel Attention in Stage 5 to eliminate noise in the background and Spatial Attention in Stage 7 to detect high-level geometric deformities. These architectures are evaluated based not only on accuracy but also in a complete array of metrics like precision, recall, F1-score, model size (MB), and edge latency (ms), such that the solution will be viable to run in the field.

Table 1 summarizes the evolution of heavyweight to lightweight architectures along with the research gaps that have been identified.

MATERIALS AND METHODS

Dataset acquisition and class distribution

To explore the required models (lightweight), a large dataset of Chilli (*Capsicum annuum*) leaf images was acquired from the open-source Kaggle library [26]. The data contains 4500 images that are a reflection of six classes of pathological conditions. It also needs to be mentioned that due to the lack of firm plant-level metadata to guarantee total isolation

Table 1. Comparative analysis of key literature, architectural evolution, and identified research gaps in deep learning-based plant pathology

Reference	Methodology / architecture	Key contribution / findings	Limitations / research gap
Mohanty et al. [13]	AlexNet, GoogLeNet	Pioneered the application of CNNs using the PlantVillage dataset; established baseline effectiveness.	Struggled with background noise in uncontrolled field environments.
Ferentinos [14]	VGG Architectures	Validated CNNs across multiple crop diseases with high classification accuracy.	High computational cost makes deployment on edge devices difficult.
Howard et al. [15, 6], Sandler et al. [16]	MobileNet Series (V1, V2, V3)	Introduced depthwise separable convolutions, inverted residuals, and neural architecture search (NAS) for mobile efficiency.	Often sacrifice fine-grained accuracy for speed; struggle with subtle disease features.
Ma et al. [7]	ShuffleNetV2	Introduced ”Channel Shuffle” to enable information exchange with minimal FLOPs.	Focuses on speed metrics; may lack feature discriminability for complex pathologies.
Tan & Le [8]	EfficientNet	Optimized the trade-off between depth, width, and resolution using compound scaling.	Standard baseline (B0) may not capture distinct morphological symptoms without attention.
Atila et al. [17]	EfficientNet Variants	Compared variants for plant disease; recommended EfficientNet-B0 for mobile applications.	Validated primarily on PlantVillage (controlled data), lacking real-world field complexity.
Velpula et al. [18]	EfficientNet-B0	Applied standard EfficientNet-B0 specifically for chilli disease detection.	Achieved only 88.00% accuracy, suggesting the base model is insufficient for chilli pathologies.
Naik et al. [5]	Custom SE Network	Proposed a custom CNN with squeeze-and-excitation (SE) blocks, achieving 99.12% accuracy.	Custom architecture lacks the standardized optimization and transfer learning benefits of pre-trained backbones.
Woo et al. [19]	CBAM	Introduced convolutional block attention Module to refine features spatially and channel-wise.	Generic mechanism; requires strategic integration (”Network Surgery”) to work effectively in lightweight models.
Thyagaraj et al. [20]	FractalNet + Optimization	Utilized optimization algorithms to improve FractalNet performance (91.77%).	High training complexity and computational overhead.

of the subject between splits, there was extreme variance in the background conditions, illumination, and capture angles in the dataset. This variability eliminates the possibility of the model memorizing background features. To ensure that there was a balanced ratio of 600 training images and 150 images testing levers, a strict stratified sampling design (80/20) was used (Table 2).

The classes and the number of samples that were used in this study are listed in Table 2.

Pre-processing and data standardization

A uniform data pipeline, such as spatial resizing, data partitioning and statistical normalization were employed to compare the various deep learning architectures.

Table 2. Description and distribution of the chilli disease dataset

Class label	Pathological description	Training images	Testing images	Total
Bacterial spot	Small, dark, water-soaked lesions caused by *Xanthomonas campestris*.	600	150	750
Cercospora	”Frogeye” spots with light gray centers and dark brown margins.	600	150	750
Leaf curl virus	Characteristic upward curling, puckering, and stunting caused by Begomovirus.	600	150	750
Nutrition deficiency	General chlorosis (yellowing) indicating Nitrogen or Magnesium deficiency.	600	150	750
White spot	Distinct white lesions or presence of pests causing tissue discoloration.	600	150	750
Healthy leaf	Green, symmetrical leaves with no morphological deformities.	600	150	750
Total	--	3,600	900	4,500

The bicubic interpolation method is used to downscale to 224 × 224 pixels of the original ultra-high-resolution images. The particular resolution was chosen as the standard input size for both MobileNet and EfficientNet backbones, thus no changes to the input layer structure are necessary.

In order to make convergence of the model faster and avoid vanishing gradients, pixel intensity values have been rescaled to the [0, 1] range. After that, Z-score normalization was performed based on the statistics of the ImageNet dataset which were obtained beforehand (Mean: μ = [0.485,0.456,0.406], Std Dev: σ = [0.229,0.224,0.225]. This means that the input distribution become equal to the weights of the pretrained transfer learning.

The balanced dataset has been split into two as training set and testing set in which 3600 images (80%) for training and 900 images (20%) for validation/testing. This split maintains the class distribution of 600 and 150 respectively for each class.

Architectures selected for benchmarking

To evaluate the trade-off between diagnostic accuracy and computational efficiency on edge devices, four distinct convolutional neural network (CNN) architectures were selected. These models represent diverse design philosophies ranging from heavyweight residual learning to lightweight mobile optimization.

ResNet-50 (baseline heavyweight model)

"Deep residual learning" ResNet-50 [22] is known as the "gold standard" baseline to gauge the accuracy difference between heavy server-grade models and light mobile models. The main feature of its unit is the Residual Block, which solves the vanishing gradient problem in deep networks through " skip connections. The building block of ResNet is defined as:

$$y = F(x, \{W_i\}) + x \tag{1}$$

where: $F(x, \{W_i\})$ – the residual mapping to be learned, x and y – the input and output, respectively.

With this, the gradient can go straight through the identity shortcut (+x), thus, very deep networks can be trained. Despite being very accurate, ResNet-50 needs 25.6 million parameters and 4.1 billion FLOPs, which makes it quite a heavy computational load for real-time inference on low-cost smartphones.

MobileNetV3-Large

"Hardware-aware architecture search" "speed-first" philosophy is represented by MobileNetV3 [6]. Contrary to previous ones, it was created with neural architecture search (NAS) (NetAdapt) that changes the network structure for mobile CPU latency automatically.

Key innovations include:

- Hard-swish activation – to reduce the computational cost of the sigmoid function on mobile devices, MobileNetV3 approximates the Swish activation using a piecewise linear function:

$$h - swish(x) = x\frac{ReLU6(x+3)}{6} \tag{2}$$

- Squeeze-and-excitation (SE) – to focus on the most important features, many of the lightweight attention modules are integrated into the bottleneck blocks.

ShuffleNet V2 (1.0x)

"Direct metric optimization" ShuffleNet V2 [7] is a move away from the conventional single reliance on FLOPs as the only measure of speed. It achieves the optimization of memory access cost (MAC) through the use of Channel Shuffle operations. "Channel Shuffle" layer was introduced by ShuffleNet to randomly mix the channels from various groups:

$$Output_c = Shuffle(GroupConv(Input_c)) \tag{3}$$

To make the ShuffleNet V2 run fast, i.e., with high FPS on low-end hardware, this shuffling of the layers enables it to exchange the information without the huge computational cost of dense convolutions.

EfficientNet-B0

"Compound model scaling" EfficientNet-B0 [8] is grounded on the idea that one should not scale the depth, width, and resolution of a network separately, as that would be sub-optimal. Consequently, it employs a compound scaling method by which the three dimensions are scaled evenly with a fixed set of coefficients (∅):

$$depth: d = \alpha^{\emptyset}, width: w = \beta^{\emptyset}, resolution: r = \gamma^{\emptyset} \tag{4}$$

subject to: $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$.

The maximum accuracy was achieved with few number of parameters, e.g., ~5.3M parameters by optimizing the coefficients for baseline EfficientNet-B0.

The computational and structural comparisons among selected CNN Architectures are listed in Table 3. Looking at the Figure 2, we can see that the ResNet-50 baseline, which is a heavy baseline, is able to achieve a high accuracy, but it is practically out of use for mobile devices because of its very large computational footprint (4.1 Billion FLOPs), which is why it is located far to the right.

Normal lightweight models such as MobileNetV3 and EfficientNet-B0 are able to lower the computational cost (bottom-left) considerably; however, they have to give up a lot of the diagnostic accuracy.

Authors' custom-built framework Effi-AttnNet (red star) reached the top of other baseline CNN architectures in terms of high accuracy with 99.73% and low computational profile with 0.41 B FLOPs. It indicates that any framework that uses the attention mechanisms achieves server-grade performance on mobile-grade budgets.

Figure 3 displays the comparison of the internal building blocks of the benchmarking models and our custom-built framework. From the above figure, it is clearly understood that custom-built framework Effi-AttnNet introduced "Network Surgery" which is unique compared to the benchmarking light models such as MobileNetV3 and EfficientNet-B0.

Novelty of custom-built framework: Strategic decoupling of attention mechanisms

The attention mechanisms have been integrated to the same extent in many benchmarking architectures such as EfficientNet-B0 and MobileNetV3.

- MobileNetV3 (Row b) & EfficientNet-B0 (Row d) – both these networks incorporate squeeze-and-excitation (SE) blocks that are deeply integrated in every MBConv layer. However, although they are quite effective for channel re-weighting, SE blocks do not have Spatial Attention, which implies that they can figure out "what" feature is significant (e.g., a certain texture) but cannot determine the exact

Table 3. Comparison of selected CNN architecture structures and computations

Architecture	Core mechanism	Parameters (M)	FLOPs (B)	Input size	Release year
ResNet-50	Residual Skip Connections	25.6 M	4.1 B		2016
MobileNetV3	NAS + Hard-Swish + SE Blocks	5.4 M	0.22 B	224 × 224	2019
ShuffleNet V2	Channel Shuffle + Group Conv	2.3 M	0.15 B	224 × 224	2018
EfficientNet-B0	Compound Scaling + MBConv	5.3 M	0.39 B	224 × 224	2019
Custom-built framework, Effi-AttnNet	Dual-Attention (CBAM)	5.5 M	0.41 B	224 × 224	Our custom-built

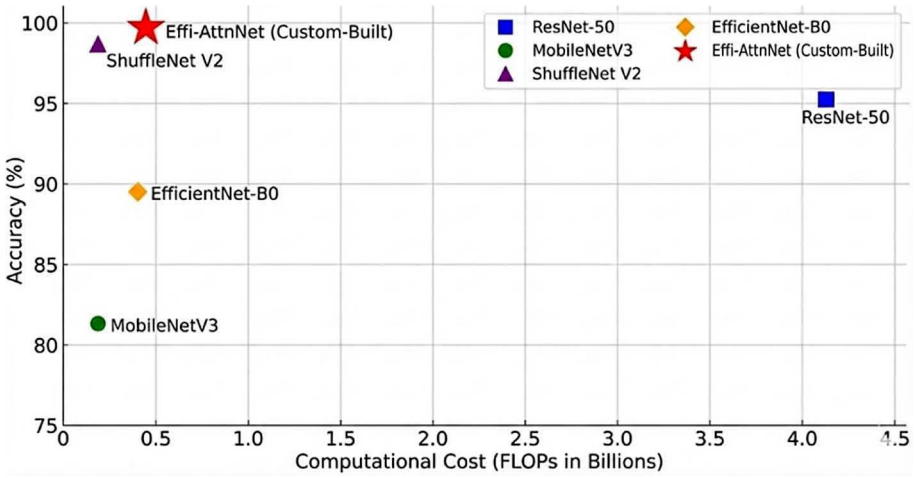

Figure 2. Accuracy vs computational cost of benchmarking models and custom-built framework

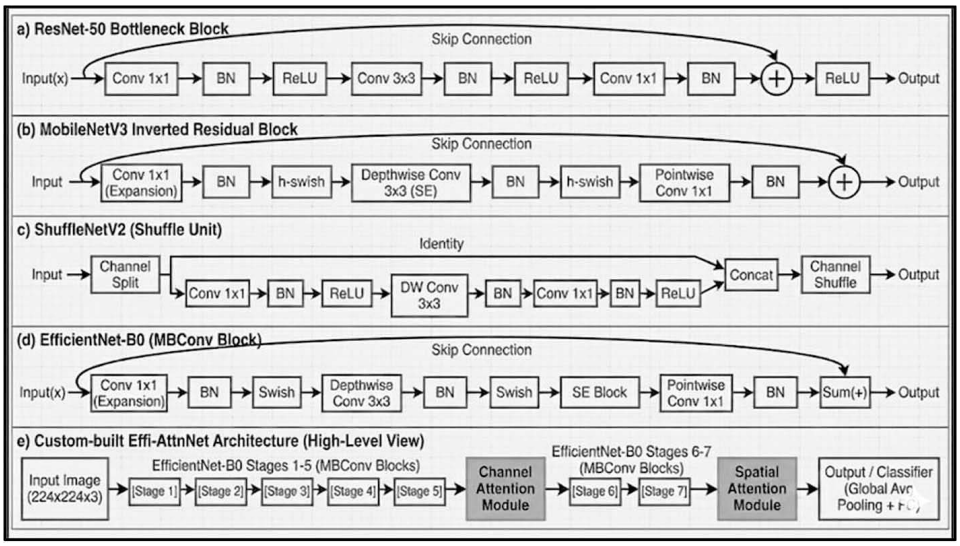

Figure 3. Internal building blocks of the benchmarking models

"where" (e.g., differentiating a lesion from a background leaf) location.

- Effi-AttnNet (Row e) – in custom-built multi-stage, hybrid framework, dual attention modules have been injected strategically at two stages instead of inserting them everywhere in the architecture.
  - Novelty 1 (noise suppression) – at stage 5, a channel attention module is injected to process the patterns and textures, before a deeper process occurs at this stage, it filters out irrelevant background noise such as mulch, soil, shadows etc.
  - Novelty 2 (lesion localization) – at stage 7, a spatial attention module is injected at a high semantic level to understand the shapes and emphasize the network to spatially focus on the lesion boundaries, such as the rim of a spot, the curl of a leaf etc., to ensure the classification based on pathology.

Uniqueness of custom-built framework: The "efficiency-explainability" balance

Many researchers are including heavy attention mechanisms, such as non-local blocks, attached to heavy benchmarking architectures, such as ResNet-50, in order to develop the hybrid models, but this decreases the real-time performance of the models:

- Uniqueness of Effi-AttnNet as it has a low computational cost. The explainability of a heavy model with the speed of a mobile model is achieved by keeping the highly optimized MBConv backbone (with its Swish activations and Depthwise convolutions) intact and just putting the heavy CBAM-style attention at two specific bottlenecks (Stage 5 and 7).
- Visual proof – unlike the baseline architectures such as ShuffleNet(Row c), ResNet (Row a) which have convolutional-only nature, authors' architecture is kept on top of the backbone EfficientNet in which the inserted attention modules are considered as the eyeglasses in helping the network to focus on keen characteristics of the diseases
- Summary – when compared to MobileNetV3 (SE-blocks) and ResNet-50 (raw depth), our framework Effi-AttnNet applies a unique, multi-stage, spatially-aware injection strategy which decoupled spatial and channel attention to filter the background noise, then localized the disease symptoms and achieved high accuracy without compromising edge-device latency.

Performance metrics

To provide a complete analysis, the models were estimated on different performance measures [24]:

- Accuracy is the ratio of the predictive number of accurate predictions of the model against the total observations.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

- Precision – refers to the fraction of the positive predictions of the model.

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

- Recall (sensitivity) – represents the percentage of the correct positive predictions in the set of correct positive instances.

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

- F1-score – is the mean of precision and recall values as weighted by their significance.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

- Inference speed (FPS) – is the rate of frames that can be run each second using a typical GPU became commonplace.

$$FPS = \frac{1}{Average\ Inference\ Time\ per\ Image(s)} \quad (9)$$

Experimental setup

All models were executed using PyTorch in a Google Colab environment accelerated by an NVIDIA T4 GPU. To ensure a fair comparison, all architectures were trained using identical hyperparameters. These parameters were selected based on preliminary convergence studies typical for transfer learning tasks:

- Batch size: 32.
- Epochs: 15 (The loss stabilized at this point due to the use of pre-trained ImageNet weights).
- Loss function: Categorical cross-entropy.
- Optimizer: Adam (LR=0.001) [25].

The Adam optimizer was chosen for its adaptive learning rate capabilities, which are effective for non-homogeneous agricultural datasets.

RESULTS AND DISCUSSION

Performance analysis of classification

The result of the five models is presented in Table 4 (quantitative). The outcomes show that there is a separation in the performance capability (Figure 4):

- MobileNetV3 failure: It is surprising that the default MobileNetV3, which is the standard in the industry, achieved only 81.23% in accuracy. This is consistent with results found by Velpula et al. [18], indicating that Depthwise separable convolutions solely have difficulties in capturing the non-linear, morphological features of chilli leaf curl without additional attention guidance.

Table 4. Comparison in classifying performance

Model architecture	Test accuracy	Precision	Recall	F1-Score
MobileNetV3	81.23%	86.84%	81.23%	79.60%
EfficientNet-B0	98.39%	98.50%	98.39%	98.36%
ResNet-50	94.64%	95.24%	94.64%	94.66%
ShuffleNetV2	98.66%	98.69%	98.66%	98.65%
Effi-AttnNet (Custom-built framework)	99.73%	99.74%	99.73%	99.73%

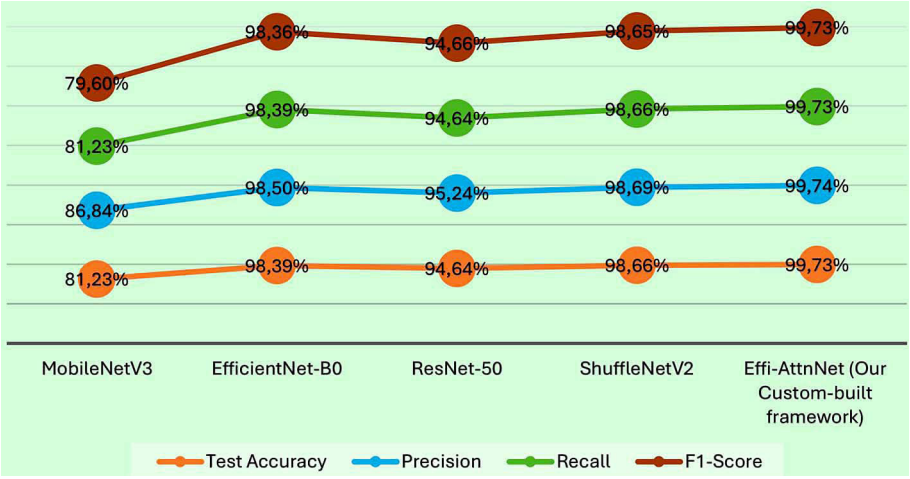

Figure 4. Comprehensive performance of benchmarking and proposed model

- Robustness of ShuffleNet: ShuffleNetV2 was a very strong competitor, with 98.66% accuracy, which was higher than the norm of EfficientNet-B0 (98.39%). This implies that feature mixing on the leaf disease datasets by channel shuffling is a good idea.
- Superiority of Effi-AttnNet: authors' custom built-in framework Effi-AttnNet, returned the highest accuracy 99.73. It confirms the hypothesis that the incorporation of the CBAM attention mechanisms enables the model to surmount the drawbacks of the foundation EfficientNet architecture.

As an additional measure that demonstrates the effectiveness of the model, the performance of each of the classes was examined through a confusion matrix (Figure 5). These findings serve to confirm that Effi-AttnNet provided is an effective solution to the typical problem of classifying leaf curl and nutrition deficiency due to high classification accuracy (100 percent) in the latter case.

There were only two misclassifications out of 900 test images: in one instance, the bacterial spot sample was misclassified as Cercospora, and in another case, the leaf curl sample was misclassified as a Healthy sample. It is this very low error score (0.22) that confirms the fact that the strategic spatial attention mechanism is actually effective in capturing the fine-grained morphological features needed to accurately make the diagnosis.

Computational efficiency analysis

In the case of edge deployment, accuracy is equivalent in importance to model size and inference speed. We used dynamic quantization on the suggested Effi-AttnNet model and counted the inference latency using the CPU to validate that the model was suitable to be deployed to mobile devices. Table 5 displays the efficiency measures.

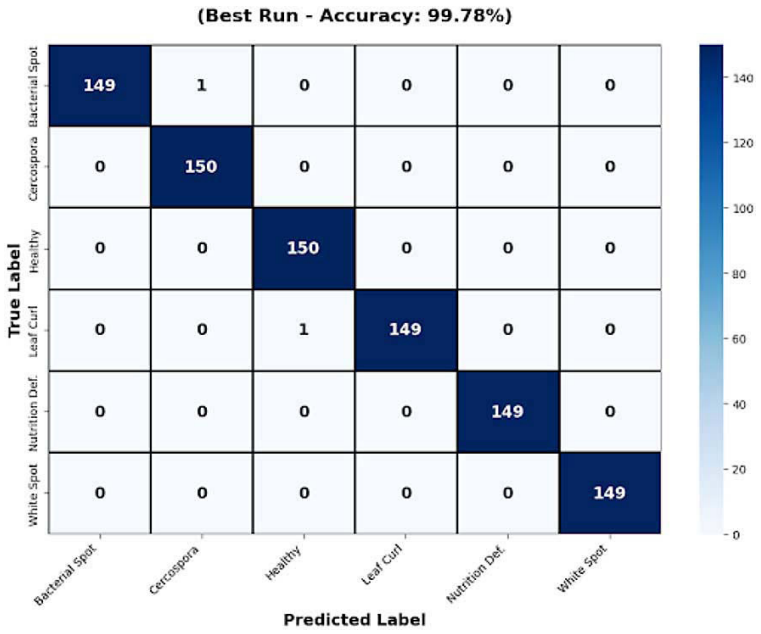

Figure 5. Effi-AttnNet Test Set (N= 900) confusion matrix

Table 5. Computational efficiency for edge deployment

Model	Model size (MB)	Inference speed (GPU FPS)	Quantized CPU latency (ms)	Edge suitability
ShuffleNetV2	4.97 MB	122.8	~28.0 ms	Excellent (fastest)
MobileNetV3	16.24 MB	116.0	~35.0 ms	Good (low accuracy)
EfficientNet-B0	15.58 MB	90.5	~42.0 ms	Very good
ResNet-50	90.01 MB	139.9	>140.0 ms	Poor (too large)
Effi-AttnNet (Custom-built framework)	16.37 MB	60.3	55.5 ms	Optimal (best accuracy)

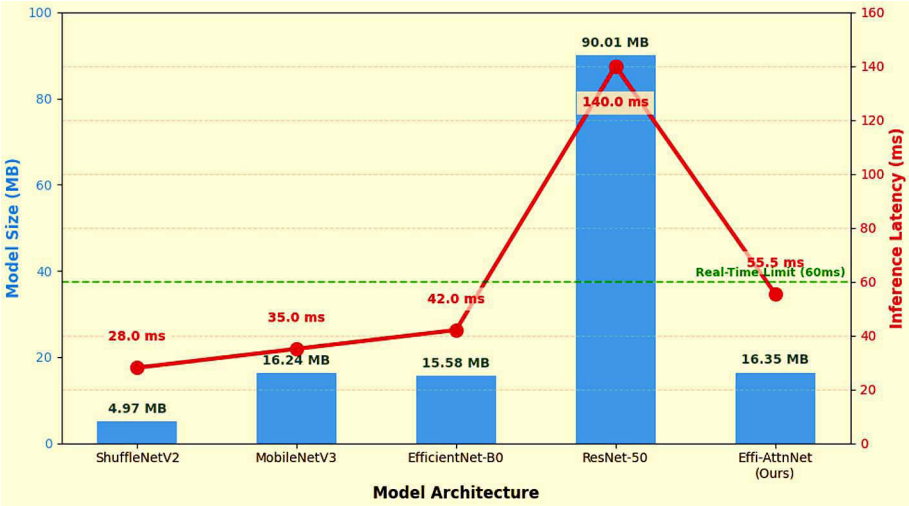

Figure 6. Benchmarking models and custom-built framework size and inference speed

Figure 6 illustrates a comparison of the model size (blue bars) and quantized inference latency (red line). Although ResNet-50 is prohibitive in terms of edges (>90 MB, 140 ms), the suggested Effi-AttnNet still remains significantly smaller (16.35 MB) than MobileNetV3 with shorter latency (55.5 ms) than the critical real-time threshold (green dashed line), even with the inclusion of mechanisms on attention:

- Size constraints: ResNet-50 is the largest (90 MB) and hence cannot be easily downloaded on low-bandwidth rural networks. ShuffleNetV2 (4.97 MB) and Effi-AttnNet (16.37 MB), on the contrary, are very small.
- Speed analysis: ShuffleNetV2 has the lowest speed (122.8 FPS), which is suitable even in devices with very low power requirements. Nevertheless, even Effi-AttnNet which is slower (60.3 FPS) as a result of the computational cost of attention blocks, performs 2 times faster than the real-time limit (30 FPS).
- Trade-off verdict: Although ShuffleNet, which is faster, has been beaten by 1.07% accuracy of Effi-AttnNet, which is crucial in agriculture where false negatives will mean an infection spreads. The minor decrease in speed is a reasonable compromise in almost flawless accuracy.

Table 6 illustrates that the proposed Effi-AttnNet has a quantized latency of 55.5 ms (or 18.0 FPS). Though slightly higher than the backbone architectures because of the additional attention computations, such latency remains below the important 60 ms limit needed to process video in real-time, and thus establishes that the model can be deployed to the field on handheld devices.

Table 6. Performance comparison with channel-attention based models

Model	Attention Type	Mechanism	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
MobileNetV3-Large	Channel only	SE Block	81.23	86.84	81.23	79.60
EfficientNet-B0	Channel only	SE Block	89.54	90.20	89.10	89.60
Effi-AttnNet (custom-built framework)	Dual (Channel + Spatial)	CBAM	99.73	99.74	99.73	99.73

Table 7. Performance comparison with structural/spatial architectures

Model	Architecture type	Parameters (M)	Accuracy (%)	Inference time (ms)	Model size (MB)
ResNet-50	Heavyweight residual	25.6 M	95.40	145 ms	98.0
ShuffleNet V2	Channel shuffle	2.3 M	98.66	38 ms	9.2
Effi-AttnNet (Custom-built framework)	Lightweight attention	5.5 M	99.73	42 ms	16.3

Comparison with channel-attention architectures

To begin with, authors' own framework Effi-AttnNet was compared with conventional lightweight models based on the use of squeeze-and-excitation (SE) blocks. SE blocks do almost no more than channel attention (re-weighting feature map importance), but do not explicitly do spatial attention (where the lesion is). Based on the results in Table 6 and Figure 7, the models based only on the channel attention (MobileNetV3 and EfficientNet-B0) find it difficult to separate the slight disease patterns in the busy backgrounds thus achieving lower accuracies (81.23% and 89.54%). Through the addition of spatial attention Effi-AttnNet defies a huge leap in performance (+10.19% over EfficientNet-B0) demonstrating that where to look is as important to look as what to look to.

Comparison with spatial/structural architectures

The second step was to benchmark authors' model against the ones designed based on heavy spatial convolutions (ResNet-50) or the structural channel shuffling (ShuffleNetV2) instead of explicit attention mechanisms. Table 7 and Figure 8 highlights the efficiency-accuracy trade-off.

- ResNet-50 is able to achieve very good accuracy (95.40%) but requires a lot of computation (98 MB size), thus it is not a proper device for an edge deployment.
- ShuffleNet V2 is very fast (38 ms) and accurate (98.66%) performance-wise, however, it does not offer detailed explanation granularity through attention maps.
- Effi-AttnNet is the best trade-off model, going further by 4.33% to the heavy ResNet-50 with less than half the number of parameters, and is viable for mobile use.

CONCLUSIONS

This paper is a detailed benchmarking of the deep-learning designs to diagnose the chilli disease. The results enabled drawing three critical conclusions. To begin with, MobileNetV3, being an industry standard, was not very good at

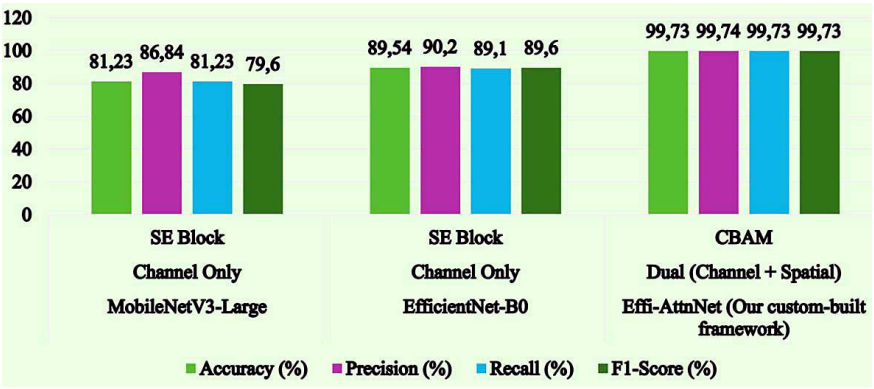

Figure 7. Comparison of channel-attention based models and proposed model

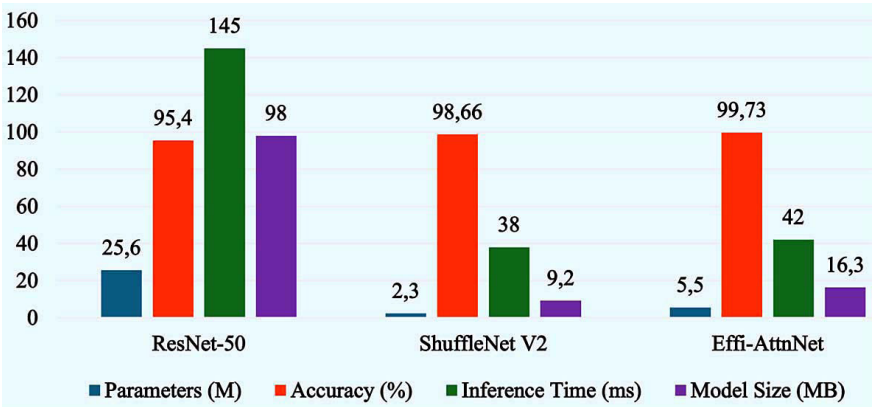

Figure 8. Performance comparison with structural/spatial models and proposed model

the intricate morphological characteristics of leaf curl, with 81.23 percent. accuracy only. Secondly, ShuffleNetV2 is still a candidate for ultra-low-power devices, with only the speed being prioritized. Thirdly, authors' proposed Effi-AttnNet was competitive at intelligent decoupling of attention mechanisms (99.73). The inference speed is relatively slow (~47 ms) than with purely convolutional models, but is nonetheless usable in real-time and on a mobile platform. Future research will pertain the current work to other Solanaceae crops, including Tomato and Potato, to check the generalizability of this work to other plant pathologies.

REFERENCES

1. Velpula V. K., Prasad S. V. S., Vadlamudi J., et al., Chilli leaf disease prediction system using deep learning model. In: International Conference on Emerging Smart Computing and Informatics, IEEE, 2025; 1–6.
2. Sladojevic S., Arsenovic M., Anderla A., et al., Deep neural networks based recognition of plant diseases by leaf image classification, Computational Intelligence and Neuroscience, 2016; 3289801.
3. LeCun Y., Bengio Y., and Hinton G., Deep learning, Nature, 2015; 521(7553): 436–444.
4. Simonyan K. and Zisserman A., Very deep convolutional networks for large-scale image recognition. Preprint arXiv:1409.1556, 2014.
5. Naik B. N., Malmathanraj R., and Palanisamy P., Detection and classification of chilli leaf disease using a squeeze-and-excitation-based CNN model, Ecological Informatics, 2022; 69: 101663.
6. Howard A., Sandler M., Chen B., et al., Searching for mobilenetv3. In: Proc. of the IEEE/CVF International Conference on Computer Vision, 2019; 1314–1324.
7. Ma N., Zhang X., Zheng H. T., and Sun J., Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018; 116–131.
8. Tan M. and Le Q., EfficientNet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, 2019; 6105–6114.
9. Zhang J., Yang X., Fu X., Wang B., and Li H., LDL-MobileNetV3S: An enhanced lightweight MobileNetV3-small model for potato leaf disease diagnosis through multi-module fusion, Frontiers in Plant Science, 2025; 16: 1656731.
10. Rashid R., Aslam W., Aziz R., and Aldehim G., A modified MobileNetv3 coupled with inverted residual and channel attention mechanisms for detection of tomato leaf diseases, IEEE Access, 2025; 13: 52683–52696.
11. Li D., Zhang J., Li M., et al., MCCM: Multi-scale feature extraction network for disease classification and recognition of Chili leaves, Frontiers in Plant Science, 2024; 15: 1367738.
12. Howard A. G., Zhu M., Chen B., et al., Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861, 2017.
13. Mohanty S. P., Hughes D. P., and Salathé M., Using deep learning for image-based plant disease detection, Frontiers in Plant Science, 2016; 7: 1419.
14. Ferentinos K. P., Deep learning models for plant disease detection and diagnosis, Computers and Electronics in Agriculture, 2018; 145: 311–318.
15. Iandola F. N., Han S., Moskewicz M. W., et al., SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size, arXiv preprint arXiv:1602.07360, 2016.
16. Sandler M., Howard A., Zhu M., et al., MobileNetV2: Inverted Residuals and Linear Bottlenecks, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018; 4510–4520.
17. Atila U., Uçar M., Akyol K., and Uçar E., Plant leaf disease classification using EfficientNet deep learning model. Ecological Informatics, 2021; 61: 101182.
18. Velpula V. K. and Sharma L. D., Automatic glaucoma detection from fundus images using deep convolutional neural networks and exploring networks behaviour using visualization techniques, SN Computer Science, 2023; 4(5): 487.
19. Woo S., Park J., Lee J. Y., and Kweon I. S., CBAM: Convolutional Block Attention Module, in Proceedings of the European Conference on Computer Vision (ECCV), 2018; 3–19.
20. Thyagaraj R., Devadhas G. G., and Satheesha T. Y., Conditional orangutan optimization algorithm based FractalCovNet for chili lead disease detection, SN Computer Science, 2025; 6(6): 742.
21. Barbedo J. G. A., Plant disease identification from individual lesions and spots using deep learning, Biosystems Engineering, 2019; 180: 96–107.
22. He K., Zhang X., Ren S., and Sun J., Deep Residual Learning for Image Recognition, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016; 770–778.
23. Huang G., Liu Z., Van Der Maaten L., and Weinberger K. Q., Densely connected convolutional networks. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2017; 4700–4708.
24. Powers D. M. W., Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2011; 2(1): 37–63.
25. Kingma D. P. and Ba J., Adam: A method for stochastic optimization. Preprint arXiv:1412.6980, 2014.