

A Lightweight Convolutional Neural Network with Neighbourhood Attention and a 100-Category Dataset for Plant Disease Detection

Rithambara Rajput*

KLE Technological University's, Dr. M. S. Sheshgiri Campus, Belagavi, 590008, India

E-mail: rithambararajput28@gmail.com

ORCID iD: <https://orcid.org/0009-0001-1981-5686>

*Corresponding Author

Suneeta Budihal

KLE Technological University, Hubballi, 580031, India

E-mail: suneeta_vb@kletech.ac.in

ORCID iD: <https://orcid.org/0000-0002-7068-2110>

Saroja Siddamal

KLE Technological University, Hubballi, 580031, India

E-mail: sarojavs@kletech.ac.in

ORCID iD: <https://orcid.org/0000-0002-9579-9980>

Dattaprasad Torse

KLE Technological University's, Dr. M. S. Sheshgiri Campus, Belagavi, 590008, India

E-mail: datorse@klescet.ac.in

ORCID iD: <https://orcid.org/0000-0002-2101-487X>

Received: 13 January, 2026; Revised: 01 March, 2026; Accepted: 28 April, 2026; Published: 08 June, 2026

Abstract: Plant disease detection is vital for agricultural sustainability and food security. While Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) have achieved high accuracy in this domain, CNNs often require millions of parameters and substantial computation. ViTs suffer from the quadratic time and space complexity of self-attention (SA), limiting their use on resource-constrained devices. Although SA is capable of modelling long-range dependencies when symptoms are dispersed, many plant diseases exhibit small, localized lesions or texture changes; therefore, Neighborhood Attention (NA) offers a more efficient and targeted alternative by focusing on nearby regions rather than the entire image.

This work proposes a custom Localized NA block implemented in TensorFlow/Keras that operates directly on CNN feature maps, bypassing patch embedding and transformer modules. A lightweight CNN is then developed by combining depth-wise separable convolutions with the proposed localized NA block. In addition, a 100-category plant disease dataset covering 16 crops is presented. The dataset is curated, class-balanced, and made publicly available to support reproducibility and encourage further research.

The proposed 9-layer CNN, with just 1.7M parameters and a size of 6.74 MB, achieved a favorable balance between accuracy, model size, and computational efficiency, compared with MobileNetV1, MobileNetV2, DenseNet121, InceptionV3, MobileViT-XXS, and EfficientViT-M0, achieving $98.97\% \pm 0.33\%$ accuracy on PlantVillage and $93.36\% \pm 0.28\%$ on the proposed dataset. The ablation study showed that the NA block improved test accuracy by approximately 2–3%, while Grad-CAM visualizations indicated more precise targeting of diseased areas in the leaf image.

Index Terms: Depth-wise Separable Convolution, Lightweight Neural Network, Linear Time and Space Complexity, Neighbourhood Attention

1. Introduction

Plant diseases threaten global agriculture; therefore, prompt and precise detection is the key to avert devastating consequences for food security and economic loss. Conventional methods of disease identification were riddled with manual inspections and were not very effective [1]. However, with the advancements in deep learning, the landscape of plant disease detection has completely changed, enabling automated, rapid and highly accurate diagnosis. At present, there exist two main architectures for the identification of plant diseases, CNNs and ViTs [2]. The CNN based architectures effectively capture spatial features, such as spots and lesions on leaves, and then scrutinize small image patches for detailed disease features while ViTs on the other hand skillfully perceive global features by studying the connection across the full image. The combination of these two techniques boosts the accuracy of the results, with CNNs pinpointing detailed symptoms and ViTs providing broader context [3].

CNNs have evolved further in complexity in the last decade in order to achieve better classification accuracy. Traditional CNNs such as AlexNet, VGG, and ResNet [4–6] exhibit impressive performance in terms of accuracy in the classification of plant diseases, but because of the higher number of parameters involved as well as the considerable computations required, they cannot be effectively applied in edge devices.

Plant disease detection using ViTs has proven effective in the field by capturing long-range dependencies. However, due to the quadratic complexity associated with the SA process, they are highly limited in real-time agriculture applications [7]. Lightweight hybrid architectures such as MobileViT, ConvNeXt, EfficientViT, and TinyViT [8–11] offer reduced computation cost. However, their transformer-style modules and scaling behavior still introduce overhead for low-resource deployment. Ultimately, the large model sizes and computing demands of standard CNNs and ViTs remain unsuitable for resource-limited settings.

Plant disease symptoms often appear as small and localized lesions. While global SA can model dispersed patterns, it computes pairwise attention across the entire image, resulting in time and space complexity growing quadratically with the number of pixels. SA increases overhead and may reduce focus on critical symptomatic regions. Localized NA restricts attention to nearby regions and aligns more closely with symptom characteristics. For a fixed neighbourhood size, NA reduces attention complexity with respect to the number of spatial locations. Existing transformer-based NA implementations [12, 13] retain heavy global components; therefore, the proposed work adopts a lightweight NA block specifically designed for plant disease detection.

This manuscript therefore presents a simple yet efficient lightweight neural network. It combines depth-wise separable convolutions with localized NA for effective local feature extraction and contextual modelling. The proposed NA is a TensorFlow/Keras adaptation optimized for convolutional feature maps and depth-wise separable CNNs, as opposed to prior Neighbourhood Attention Transformer (NAT) [14–16] implementations designed for large-scale transformer architectures. By combining depth-wise separable convolutions with NA, the proposed model captures disease symptoms at multiple scales while remaining lightweight.

The primary contributions of this paper include:

1. Custom NA Block: A compact NA block is developed in TensorFlow/Keras. It is tailored to efficiently work with depth-wise separable CNNs. Differing from previous NAT implementations devised for massive transformer architectures, the proposed NA is a lightweight model with reduced time-space complexity. As a result, it is more apt for lightweight plant disease detection.
2. 100-Category Plant Disease Dataset: This manuscript introduces a 100-category plant disease dataset covering 16 different plant species. The training subset was augmented to contain 100,000 RGB images, while the validation and test subsets were kept unchanged for unbiased evaluation. All images were resized to 256×256 pixels. This dataset serves as a valuable resource for plant health diagnostics research.
3. Custom Lightweight CNN Model Development: This work introduces a customized lightweight CNN that incorporates depth-wise separable convolutions with the NA module. The architecture consists of nine depth-wise separable convolutional layers and an NA layer. The finalized model is compact, with a size of 6.74 MB and only 1.7 million parameters.
4. Performance Assessment: The model is trained and evaluated on both the PlantVillage and 100-category plant disease datasets. Its performance is assessed using classification metrics, ablation analysis, computational complexity analysis, Grad-CAM visualization, occlusion sensitivity analysis, and comparison with selected lightweight CNN and CNN-ViT models. These evaluations demonstrate the effectiveness and interpretability of the proposed model for plant disease classification.

2. Related Work

Deep learning algorithms for plant disease detection have progressed from CNN-based methods to the modern ViT models. Recently, hybrid CNN-ViT lightweight models have demonstrated high accuracy for real-time plant disease detection. This section will categorize the studies into three groups: (1) CNN-based approaches, (2) ViT-based approaches, and (3) recent lightweight ViTs for plant disease detection.

2.1. CNN-based methods

CNN-based structures like AlexNet [17], GoogleNet [18,19], LeNet [20], VGG [21–23], ResNet, and Inception were some of the first models employed in plant disease diagnosis, exhibiting superior classification precision in various banana, apple, cassava, and PlantVillage datasets [24–27]. Nevertheless, their real-world implementation is hindered by several limitations related to architectural and computational considerations, including small gray-scale input images in LeNet, bigger kernels in AlexNet, 130 million parameters and a 500 MB model size in VGG, and heavy computation requirements for more complex architectures like ResNet and Inception [28, 29]. The need for more efficient solutions led investigators to look into light-weight frameworks, which include MobileNet and compact CNN. The effectiveness of MobileNet has been proven through multiple studies with competitive performance results in the bean leaf, tomato-potato-bell pepper, and coffee leaf disease datasets [30–33] whereas small CNNs like a nine-layer CNN in PlantVillage, a multilayer CNN in mango leaves, LeafNet for tea diseases, DICNN for grape diseases, and five-layer DCNN in 26 classes proved that light-weight CNNs could still exhibit impressive performance capabilities [34–38].

2.2. ViT-based methods

The recent trend in plant disease detection combines ViTs with CNNs to improve both accuracy and efficiency. For example, the PlantXViT architecture merges VGG16 and Inception v7 for local features with transformers for global representation, achieving strong results on apple, maize, and rice datasets [39]. In another case, four ViT models were tested against four CNN models for maize leaf disease classification, with ViT-B/16 achieving the highest accuracy of 94.51% [40]. Similarly, six ViT variations were evaluated for tomato leaf disease recognition, with Compact ViT reaching 97% accuracy, 97% precision, and 96% recall, while Mobile ViT performed worst [41]. A lightweight CNN with soft attention and depth-wise separable convolutions achieved 99.5% training accuracy with just 221,594 trainable parameters and a 2.5 MB size for tomato diseases [42].

2.3. Lightweight CNN-ViT based methods

Table 1. Summary of related work in plant disease detection, highlighting the strengths, limitations, and research gaps addressed by the proposed approach.

Method category	Representative studies	Main strengths	Key limitations	Research gap addressed in this work
Conventional CNN-based models	AlexNet, GoogleNet, LeNet, VGG, ResNet, and Inception-based plant disease models [17 - 29]	Strong local feature extraction; high accuracy on controlled or benchmark datasets such as PlantVillage	Large parameter count, high storage requirement, and significant computational cost; limited suitability for edge or real-time deployment	Need for a compact CNN model with lower computational cost and smaller model size
Lightweight CNN-based models	MobileNet-based models, compact CNNs, MCNN, LeafNet, DICNN, and 5-layer/9-layer CNN models [30 - 38]	Reduced model size and faster inference compared with large CNN backbones	May lose contextual information needed to distinguish visually similar diseases; many studies are limited to specific crops or fewer disease classes	Need for a lightweight model that preserves local feature extraction while improving contextual representation
ViT-based models	PlantXViT, ViT-B/16, Compact ViT, and other ViT variants for maize, tomato, apple, and rice disease detection [39 - 41]	Effective global dependency modelling; useful for capturing broader image-level relationships	Global self-attention has quadratic time and memory complexity; often requires more data and computational resources	Need for an attention mechanism that avoids full-image self-attention while still capturing useful context
Attention-enhanced CNN models	Soft-attention CNN with depth-wise separable convolutions [42]	Improves feature focus while maintaining a compact architecture	Attention is not explicitly neighbourhood-based; evaluation is often crop-specific	Need for localized attention that directly focuses on neighbouring disease-relevant regions
Lightweight CNN-ViT hybrid models	Enhanced MobileViT, MobilePlantViT, Efficient-ViT B0Net, MobileViT-DAP, ViT-RoT, CNN→ViT, and integrated CNN-ViT models [43 - 49]	Combine CNN local feature extraction with transformer-based contextual modelling; often achieve high accuracy with fewer parameters	Transformer-style modules may still introduce computational overhead; many models are validated on limited crop-specific datasets; deployment complexity remains a concern	Need for a simpler lightweight hybrid design suitable for multi-crop, multi-disease classification
Proposed approach	Depth-wise separable CNN with localized Neighbourhood Attention	Combines parameter-efficient convolution with local contextual attention; focuses on nearby regions aligned with localized disease symptoms; compact model size	Requires further external field validation and robustness testing under uncontrolled real-world conditions	Provides a lightweight NA-based CNN designed for 100-category plant disease detection with improved balance between accuracy, compactness, and efficiency

Building on these developments, recent studies in plant health diagnostics have increasingly adopted lightweight CNNs, ViTs, and hybrid CNN-ViT architectures to balance accuracy with deployability. Enhanced MobileViT variants with Efficient Channel Attention (ECA) and multi-scale retinal enhancement have achieved high weed-recognition

performance (F1 $\approx$ 98.5%) on mobile devices [43], while MobilePlantViT, using depth-wise separable and group convolutions with CBAM, reached up to 99% accuracy with only 0.69M parameters [44]. Efficient-ViT B0Net combines EfficientNet-B0's local feature extraction with ViT's global modeling to attain 99.13% accuracy for rice leaves [45], and MobileViT-DAP integrates dual attention (CA, ECA) and PoolFormer blocks to classify six rice diseases at 99.61% with just 0.75M parameters [46]. Broader benchmarking, such as ViT-RoT for tomato leaves, highlights the accuracy of ConvNeXt- and Swin-based designs but also the resource sensitivity of pure ViTs [47]. Domain-specific hybrids, including CNN $\rightarrow$ ViT for mulberry leaf disease (95.6% accuracy) [48] and an integrated CNN-ViT for maize leaves (99.15% combined-dataset accuracy, 95.9% cross-dataset) [49], confirm the advantage of pairing CNN locality with Transformer-based context modeling. Table 1. presents a summary of the related work in plant disease detection, highlighting the major strengths, limitations, and research gaps in existing approaches.

2.4. Observation and motivation for this work

Despite advancements in lightweight architectures, existing solutions face specific limitations. When using transformer-based models for plant disease diagnosis, the high computing requirements of MobileViT and TinyViT, the parameter counts in ConvNeXt and the scaling complexity in EfficientViT are major concerns. The use of these models in agricultural contexts, where both computing and power are severely limited, is impeded.

The proposed NA addresses this by focusing attention on local neighbourhoods. It simplifies complexity to a linear form, which is consistent with the spatial correlation of plant disease symptoms. Combining NA with parameter-efficient depth-wise separable convolutions, the suggested architecture manages to find a balance between compactness and representation. NA preserves spatial locality and scales attention efficiently, making it suitable for identifying subtle disease cues in complex leaf images while maintaining a smaller and lightweight model. As a result, this synergy has fewer parameters, a smaller size, linear complexity, and is subsequently more accurate and efficient.

3. Materials and Methods

This section presents the overall research workflow. It provides details of the customized 100-category plant disease dataset and reviews depth-wise separable convolutions. It then introduces the NA mechanism and its implementation details, followed by the implementation of the proposed lightweight model, and concludes with its training process, experimental results, and evaluation against benchmark models.

3.1. Research Workflow

Fig. 1. depicts the complete research pipeline. It starts with data collection, preprocessing, and creating a balanced 100-category plant disease dataset. The primary contribution of this study is the creation of a custom lightweight CNN combined with a customized Localized NA block created specifically for convolutional feature maps. It differs from conventional NAT designs, which were originally designed for transformers. The fundamental architectural novelty of the study is the integration of depth-wise separable convolutions with a NA mechanism. It permits linear complexity and extremely efficient local feature modeling.

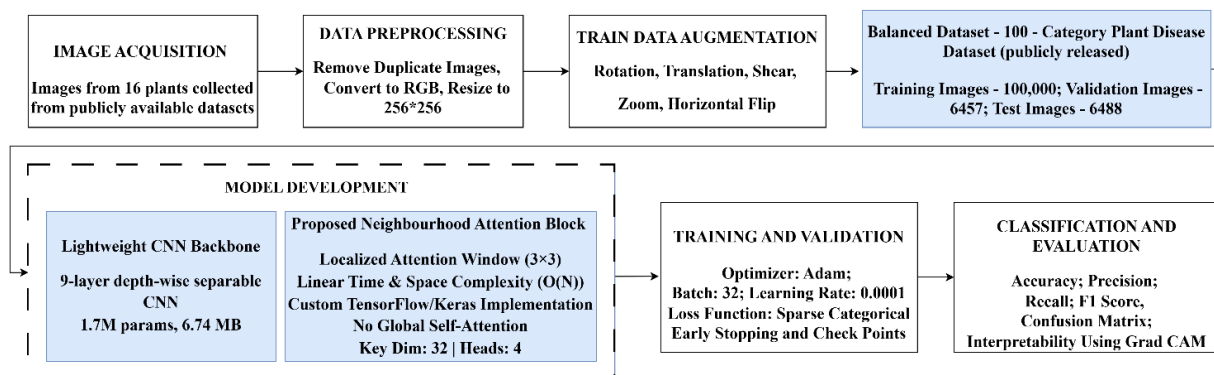


Fig. 1. Block diagram showing the process from image acquisition and augmentation to dataset balancing, lightweight neural network development with NA, training, testing, and Grad-CAM visualization

The proposed model is trained and benchmarked on both the PlantVillage dataset and the newly curated dataset to validate its robustness across diverse imaging conditions. The workflow concludes with Grad-CAM-based interpretability, which verifies the technical effectiveness of the NA block by showing its ability to consistently highlight disease-affected regions. This workflow provides a systematic pipeline for developing, evaluating, and interpreting an efficient plant disease detection model.

3.2. Customized 100-Category Plant Disease Dataset

The main limitations of PlantVillage datasets are the lack of real-world leaf images and consistent backgrounds. To address these challenges, a new 100-category plant disease dataset was created. It combines images from open-source databases for 16 plants with 100 categories. The 16 major crops are: apples, bananas, coffee, corn, cotton, eggplant, grapes, guava, mango, okra, potato, rice, sugarcane, tea, tomato, and wheat. Table 2. summarizes the source datasets along with their corresponding links and usage licenses to ensure transparency and reproducibility.

The dataset was prepared using the following steps:

1. Image collection: Images were collected from various public datasets of 16 plant species and organized into 100 plant disease categories. The original raw dataset contains 56,477 images.
2. Manual label validation: All images were manually checked for label consistency. Images that were corrupted, irrelevant, unreadable or misassigned to a class were removed. All sources were standardized in class names.
3. Duplicate removal: A perceptual hashing function was used to detect and remove duplicate or near-duplicate images. After this step, the dataset was reduced from 56,477 images to about 43,040 unique images.
4. Dataset Splitting: The cleaned dataset was split into training, validation and test in a 70:15:15 ratio before augmentation. This prevented showing augmented versions of the same image in different subsets. The resulting split contained 30,095 training images, 6,457 validation images, and 6,488 test images before augmentation.
5. Training set augmentation: Augmentation was applied only to the training set. No augmentation was performed on the validation and test sets. Augmentation methods such as rotation ($\pm 20^\circ$), translation (0.2 horizontally and vertically), shear (0.2), zoom (0.2), random horizontal flips, and fill mode nearest were performed to improve class balance and training diversity. For classes with very few images, we applied heavier augmentation to mitigate the class imbalance.
6. Image pre-processing: Images were resized to 256×256 pixels and converted to RGB format.
7. Final dataset: The augmented training set included 100,000 images, while the validation and test sets contained 6,457 and 6,488 images, respectively.

Table 2. List of plants and corresponding dataset links

	Plant	Dataset Link	License
1.	Apple	https://www.kaggle.com/datasets/nirmalsankalana/apple-tree-leaf-disease-dataset	CC0: Public Domain
2.	Banana	https://www.kaggle.com/datasets/sujaykapadnis/banana-disease-recognition-dataset https://www.kaggle.com/datasets/shifatearman/bananaalsd https://www.kaggle.com/datasets/warcoder/nutrient-deficient-banana-plant-leaves	CC BY 4.0
3.	Coffee	https://www.kaggle.com/datasets/mohammedzwaughfa/coffee-leaf-disease-dataset	MIT
4.	Corn	https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset	Data files © Original Authors
5.	Cotton	https://www.kaggle.com/datasets/ataher/cotton-leaf-disease-dataset	-
6.	Eggplant	https://www.kaggle.com/datasets/sujaykapadnis/eggplant-disease-recognition-dataset	CC BY 4.0
7.	Grapes	https://www.kaggle.com/datasets/rm1000/grape-disease-dataset-original	CC0
8.	Guava	https://www.kaggle.com/datasets/omkarmanohardalvi/guava-disease-dataset-4-types	-
9.	Mango	https://www.kaggle.com/datasets/aryashah2k/mango-leaf-disease-dataset	CC BY-NC 4.0
10.	Okra	https://www.kaggle.com/datasets/manojgadde/yellow-vein-mosaic-disease	Data files © Original Authors
11.	Potato	https://www.kaggle.com/datasets/rizwan123456789/potato-disease-leaf-datasetpld	DbCL v1.0
12.	Rice	https://www.kaggle.com/datasets/vbookshelf/rice-leaf-diseases https://www.kaggle.com/datasets/guy007/nutrientdeficiencysymptomsinrice	-
13.	Sugarcane	https://www.kaggle.com/datasets/nirmalsankalana/sugarcane-leaf-disease-dataset	CC0
14.	Tea	https://www.kaggle.com/datasets/shashwatwork/identifying-disease-in-tea-leafs	CC BY-SA 4.0
15.	Tomato	https://www.kaggle.com/datasets/cookiefinder/tomato-disease-multiple-sources	CC0
16.	Wheat	https://www.kaggle.com/datasets/amankumar2004/wheat-disease-dataset	MIT

Table 3. details the dataset, listing each category along with the number of images before augmentation in the 100-category plant disease dataset.

Table 3. Plant names, categories, and image counts in the 100-category plant disease dataset (before augmentation).

S.No	Category	Train	Val	Test	S.No	Category	Train	Val	Test
------	----------	-------	-----	------	------	----------	-------	-----	------

A Lightweight Convolutional Neural Network with Neighbourhood Attention and a 100-Category Dataset for Plant Disease Detection

1.	Apple Alternia Spot	194	42	42	51.	Guava Mummification	115	25	25
2.	Apple Brown Spot	149	32	33	52.	Guava Rust	116	25	26
3.	Apple Gray Spot	271	58	59	53.	Mango Anthracnose	334	72	72
4.	Apple Healthy	286	61	62	54.	Mango Bacterial Canker	345	74	74
5.	Apple Rust	240	52	52	55.	Mango Die Back	170	36	37
6.	Banana Black Sigatoka Disease	46	10	11	56.	Mango Gall Midge	338	72	73
7.	Banana Boron Deficient	70	15	15	57.	Mango Healthy	345	74	74
8.	Banana Bract Mosaic Virus	35	7	8	58.	Mango Powdery Mildew	343	74	74
9.	Banana Calcium Deficient	296	64	64	59.	Mango Sooty Mould	347	75	75
10.	Banana Cordana	113	24	25	60.	Mango Weevil	347	75	75
11.	Banana Healthy Leaf	60	13	13	61.	Okra Healthy	619	133	133
12.	Banana Insect Pest Disease	60	13	13	62.	Okra Yellow Vien Mosaic Disease	491	105	106
13.	Banana Iron Deficient	58	13	13	63.	Potato Early Blight	700	150	150
14.	Banana Magnesium Deficient	103	22	23	64.	Potato Healthy	555	119	120
15.	Banana Manganese Deficient	16	4	4	65.	Potato Late Blight	700	150	150
16.	Banana Moko Disease	38	8	9	66.	Rice Bacterial Blight	28	6	6
17.	Banana Panama Disease	28	6	7	67.	Rice Brown Spot	28	6	6
18.	Banana Potassium Deficient	121	26	26	68.	Rice Leaf Smut	28	6	6
19.	Banana Pestalotiopsis	141	30	31	69.	Rice Nitrogen Deficient	305	66	66
20.	Banana Sulphur Deficient	508	109	110	70.	Rice Phosphorous Deficient	231	50	50
21.	Banana Yellow Sigatoka Disease	16	3	4	71.	Rice Potassium Deficient	266	57	58
22.	Banana Zinc Deficient	275	59	60	72.	Sugarcane Healthy	333	71	72
23.	Coffee Healthy	280	60	60	73.	Sugarcane Mosaic	263	56	57
24.	Coffee Miner	251	55	54	74.	Sugarcane Red Rot	362	78	78
25.	Coffee Phoma	268	58	58	75.	Sugarcane Rust	312	67	68
26.	Coffee Rust	280	60	60	76.	Sugarcane Yellow	345	74	75
27.	Corn Blight	700	150	150	77.	Tea Algal Leaf	79	17	17
28.	Corn Common Rust	700	150	150	78.	Tea Anthracnose	70	15	15
29.	Corn Gray Leaf	399	86	86	79.	Tea Bird Eye Spot	69	15	15
30.	Corn Healthy	700	150	150	80.	Tea Brown Blight	79	17	17
31.	Cotton Aphids	260	56	56	81.	Tea Gray Blight	68	15	15
32.	Cotton Army Worm	262	56	57	82.	Tea Healthy	51	11	12
33.	Cotton Bacterial Blight	259	56	56	83.	Tea Red Leaf	100	21	22
34.	Cotton Healthy	259	56	56	84.	Tea White Spot	99	21	22
35.	Cotton Powdery Mildew	259	56	56	85.	Tomato Bacterial Spot	700	150	150
36.	Cotton Target Spot	253	54	55	86.	Tomato Early Blight	700	150	150
37.	Egg-plant Healthy	140	30	30	87.	Tomato Healthy	700	150	150
38.	Egg-plant Insect pest	140	30	30	88.	Tomato Late Blight	700	150	150
39.	Egg-plant Leaf Spot	140	30	30	89.	Tomato Leaf Mold	700	150	150
40.	Egg-plant Milt	140	30	30	90.	Tomato Mosaic	700	150	150
41.	Egg-plant Mosaic Virus	140	30	30	91.	Tomato Powdery mildew	560	120	121
42.	Egg-plant Small Leaf	140	30	30	92.	Tomato Septoria Leaf Spot	700	150	150
43.	Egg-plant White Mold	140	30	30	93.	Tomato Spider Mites	700	150	150
44.	Grapes Black Rot	330	71	71	94.	Tomato Target Spot	700	150	150
45.	Grapes ECSA	336	72	72	95.	Tomato Yellow Curl	700	150	150
46.	Grapes Healthy	296	63	64	96.	Wheat Brown Rust	623	133	134
47.	Grapes Leaf Blight	301	64	65	97.	Wheat Healthy	700	150	150
48.	Guava Canker	116	25	25	98.	Wheat Loose Smut	177	38	38
49.	Guava Dot	74	16	16	99.	Wheat Septoria	196	42	42
50.	Guava Healthy	541	116	116	100.	Wheat Yellow Rust	630	135	135
						Total	30,095	6457	6488



Fig. 2. Sample images from 100 Category Plant Disease Dataset. The displayed images are arranged in rows of ten, ordered from left to right according to their corresponding category names

Fig. 2. illustrates sample images from the 100-category dataset, highlighting its diversity in backgrounds, lighting, and acquisition conditions. This variability addresses limitations of the PlantVillage dataset, which is predominantly captured under controlled environments.

3.3. Depth-wise Separable Convolutions

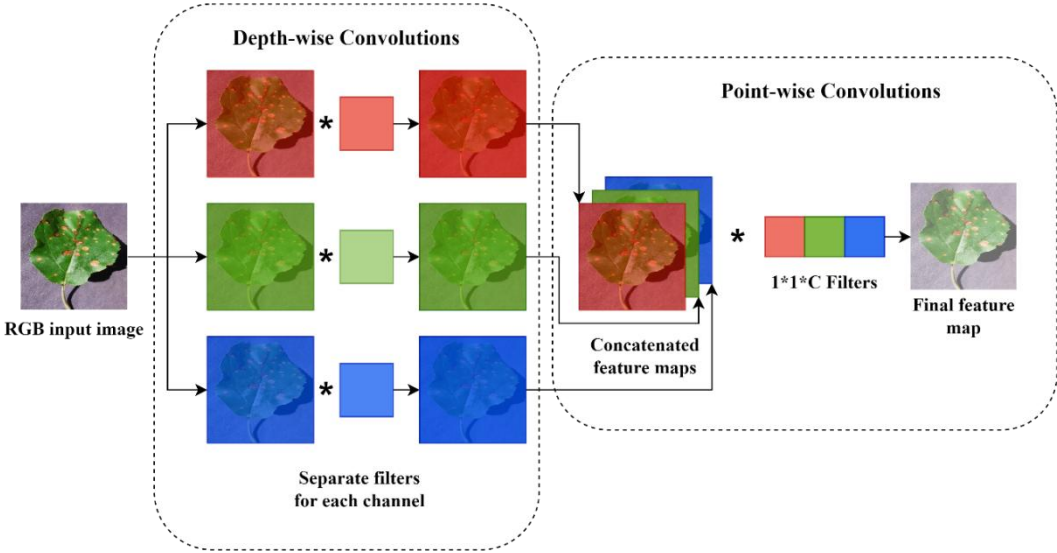


Fig. 3. Illustration of Depth-Wise Separable Convolution. Each input channel is convolved independently using depth-wise filters, and the resulting feature maps are merged through a point-wise (1×1) convolution to produce the final output. This operation significantly reduces computation while preserving spatial information

Depth-wise separable convolutions represent a significant improvement in convolutional neural networks. They separate the traditional three-dimensional convolution into two steps: depth-wise and point-wise. Depth-wise convolution applies a two-dimensional operation to each channel, which greatly reduces the number of parameters. This allows the network to extract spatial features independently. Following this, point-wise convolution uses a 1x1 kernel to combine the results of the depth-wise operation across all channels, thus reducing computational needs. Fig. 3. illustrates the architecture of depth-wise separable convolutions, highlighting the use of depth-wise and point-wise convolutions.

In depth-wise convolution, each filter is applied separately to each input channel, rather than across all channels at once. This implies that, C_{in} depth-wise filters are used for C_{in} input channels. The number of parameters for this operation is calculated as given in equation (1).

$$P=k*k*C_{in} \quad (1)$$

The total multiplication count for depth-wise convolution is described in equation (2).

$$M=H*W*k*k*C_{in} \quad (2)$$

Similarly, point-wise convolution, uses a 1*1 kernel, to merge the results of the depth-wise convolution across channels. Each point-wise filter has dimensions $1*1*C_{in}$, resulting in parameters given in equation (3).

$$P=C_{in}*C_{out} \quad (3)$$

The number of multiplications for point-wise convolution is given in equation (4).

$$M=H*W*C_{in}*C_{out} \quad (4)$$

When combined, these two processes allow for efficient feature extraction while drastically lowering the computational cost when compared to conventional convolutions. A summary of the parameters and multiplication count for standard and depth-wise separable convolutions is shown in Table 4.

Table 4. Equations summarizing the parameters and multiplication count for standard and depth-wise separable convolutions

	Number of Parameters	Number of Multiplications
Standard Convolutions	$k*k*C_{in}*C_{out}$	$H*W*k*k*C_{in}*C_{out}$
Depth-wise Convolutions	$k*k*C_{in}$	$H*W*k*k*C_{in}$
Point-wise Convolutions	$C_{in}*C_{out}$	$H*W*C_{in}*C_{out}$
Depth-wise Separable Convolutions	$(k*k*C_{in})+(C_{in}*C_{out})$	$H*W*C_{in}*[k^2+C_{out}]$

Thus, compared to standard convolutions, depth-wise separable convolutions greatly reduce parameters and multiplications, increasing efficiency and maintaining output quality. Therefore, they are suitable for lightweight models in environments with limited resources.

3.4. Introduction to NA

NA is an effective attention mechanism developed to reduce computational costs associated with traditional SA. SA examines attention at all pixels across the image, while NA concentrates on the neighbouring pixels of each query pixel according to a predetermined window size, instead of the whole image. Locally focused attention increases efficiency because it decreases the size of attention [50, 51].

Moreover, NA uses a sliding window approach, in which the neighbouring pixel window traverses the image, allowing pixels to have dynamic interactions across the pixels in the window [52]. Comparatively, NA operates with linear time and space complexity with respect to the image size, which is an improvement over the conventional SA, which has quadratic complexity.

The convergence of localized, efficient, and neighborhood interactions means that NA is particularly suited for plant disease identification, which relies on detecting fine-grained variations in local texture as well as localized and nuanced disease symptoms. NA utilizes relevant local context to improve robustness of feature representation and classification accuracy but maintains the linear time and space complexity needed to analyze collections of agricultural image datasets at scale and with high resolution. Fig. 4. visualizes both SA and NA approaches on a leaf image. The red pixel provides the query for the SA, which calculates attention across the entire image; whereas the NA focuses on the query pixel, showing a local window around it to calculate attention weights.

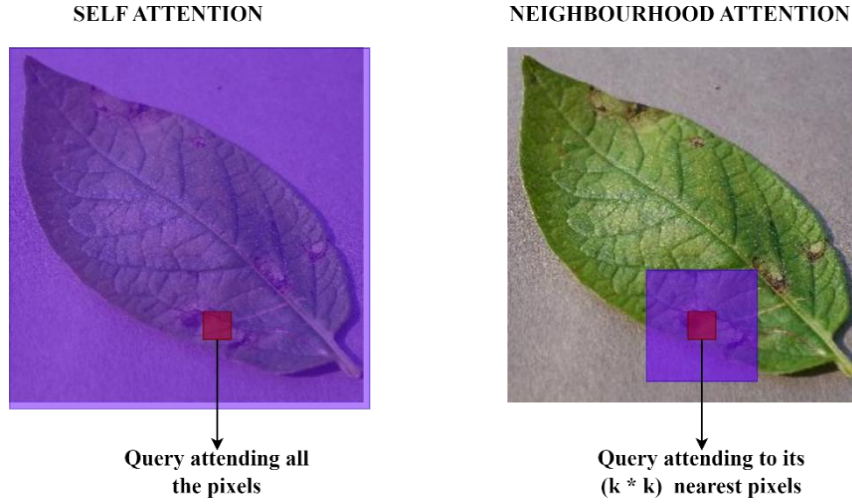


Fig. 4. In SA (left), each query attends to every pixel in the image, resulting in quadratic time and memory complexity $O(N^2)$. In contrast, NA (right) restricts each query to its local $k \times k$ neighbourhood, reducing computations to linear complexity $O(N)$.

3.4.1. Equations Describing NA:

Consider an input matrix $x \in \mathbb{R}^{N \times d}$ (where N is the number of tokens or pixels, and d signifies the dimensionality of each token), NA calculates the attention for every token by utilizing its query, key, and value projections. The matrices for query, key, and value are denoted as Q , K , V respectively.

The attention weights for the i^{th} input token, using a neighbourhood size of $k \times k$ are determined by computing the dot product of the i^{th} query Q_i and its corresponding neighbouring keys $K_{\rho_j(i)}$ (where $\rho_j(i)$ denotes the nearest j^{th} neighbour of pixel i) is described in equation (5).

$$A_i^k = \begin{bmatrix} Q_i K_{\rho_1(i)}^T \\ Q_i K_{\rho_2(i)}^T \\ \vdots \\ Q_i K_{\rho_k(i)}^T \end{bmatrix} \quad (5)$$

Neighbouring Values: The matrix of neighbouring values V_i^k is formed by utilizing the projections of the nearest neighbouring values $V_{\rho_j(i)}$ and can be written as given in equation (6).

$$V_i^k = [V_{\rho_1(i)}^T, V_{\rho_2(i)}^T, \dots, V_{\rho_k(i)}^T]^T \quad (6)$$

NA Output: The final attention output corresponding to the i^{th} token is derived as given in equation (7).

$$NA_k(i) = \text{softmax}\left(\frac{A_i^k}{\sqrt{d}}\right) V_i^k \quad (7)$$

This equation guarantees that attention weights total 1 for calculating the weighted sum of neighbouring values.

3.4.2. Linear Time and Space Complexity of NA

In SA, every token interacts with all other tokens, resulting in a time complexity of $O(N^2 d)$, where N represents the number of tokens (or pixels in case of image) and d denotes the embedding dimension. This arises from the necessity to calculate dot products for each token pair. The space complexity is $O(N^2 + Nd)$, with $O(N^2)$, allocated for the attention scores (attention matrix) and $O(Nd)$, designated for storing queries, keys, and values. Therefore, the time and space complexity for SA increases quadratically with respect to N .

In NA, each token (or pixel) only attends to a fixed-size local neighbourhood of size k . This reduces the time complexity to $O(Nkd)$ where $k \ll N$ is the neighbourhood size. The space complexity is also reduced to $O(N(k+d))$, where $O(Nk)$ is for the local attention scores and $O(Nd)$ for storing queries, keys, and values. This linear dependence on N makes NA more efficient than SA, especially for large inputs.

From Table 5. SA's scaling of time and space, being quadratic, leads to considerable computational expense, while by using a fixed-size NA is able to decrease complexity and produce a much more efficient method to handle larger inputs.

Table 5. Complexity comparison between SA and NA mechanisms.

	SA	NA
Number of Pixels (N)	N	N
Neighbourhood Size (k)	Global (all pixels)	k
Time Complexity	$O(N^2d)$	$O(Nkd)$
Space Complexity	$O(N^2+Nd)$	$O(N*(k+d))$

Thus, the NA is an effective alternative to the global SA, reducing both complexity and memory usage by emphasizing local interactions.

3.4.3. Implementation Details of NA

In this work, the NA block is implemented using TensorFlow/Keras, and takes advantage of built-in MultiHeadAttention operations limited to small spatial patches. The NA block substantially improves the overall model performance as it dynamically extracts neighbourhoods around each pixel to produce attention maps focusing on appropriate contextual information. As shown in Fig. 5., we first define the key elements are:

- Window size or Neighbourhood Size: $k \times 3$
- Number of attention heads: 4
- Key/Query dimension: 32

These parameters dictate how the attention mechanism operates within the local context of the image. In order to accommodate the edge pixels, dynamically pad this input tensor using tf.pad the appropriate padding size would be half of the kernel. In this way, every pixel will have access to its neighbors, even at the edges. The main operation for extracting neighborhoods is done using tf.image.extract_patches function; this operation slides a neighbourhood window of the given size over the padded image. The output of this function will be in the $(batch_size, height, width, kernel_size \times kernel_size, channels)$ format, where every neighborhood includes all of the neighborhoods for each pixel. After extraction, neighborhoods are reshaped to prepare them for attention computation. The queries are reshaped into a 2D tensor with dimensions $(batch_size, height \times width, channels)$, while the keys and values derived from the neighbourhoods are reshaped to $(batch_size \times height \times width, kernel_size \times kernel_size, channels)$.

Utilizing TensorFlow's MultiHeadAttention layer, the attention mechanism is processed with the queries and reshaped keys and values. The result of the attention function will have the same dimensions as the original input $(batch_size, height, width, channels)$, returning contextual information from neighboring pixels back into the primary feature representation. This local attention allows the model to learn more robust features, which improves classification performance on plant disease detection tasks.

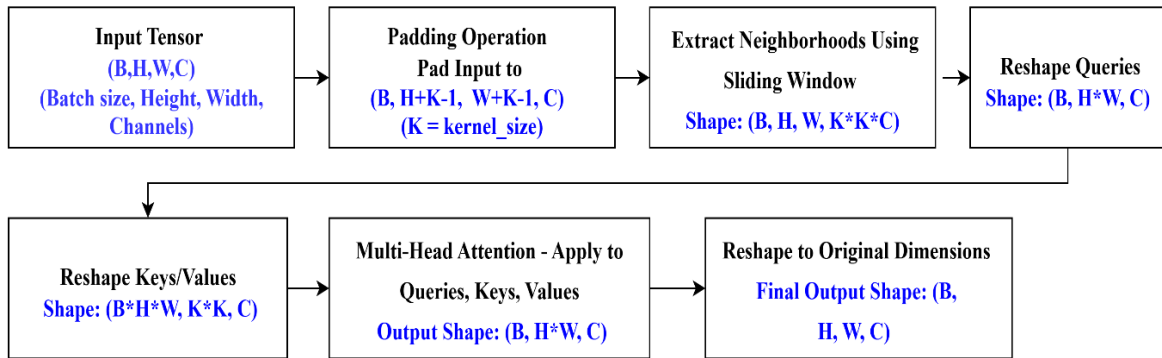


Fig. 5. Processing pipeline of the proposed Localized NA block, showing padding, neighborhood extraction, tensor reshaping, multi-head attention, and reconstruction of the output feature map.

The internal workflow of the NA block is illustrated in Fig. 5. and described as follows:

1. **Input Tensor:** Receives feature maps from the previous layer with shape (B,H,W,C) where (B) is the batch size, (H) and (W) are spatial dimensions, and (C) is the number of channels.
2. **Padding Operation:** Zero-padding is applied so that sliding windows can be centered on border pixels, resulting in shape $(B,H+K-1,W+K-1,C)$

3. **Neighbourhood Extraction:** Local spatial patches of size $(K \times K)$ are extracted for each spatial position using a sliding window, producing $(B, H, W, K^2 \times C)$
4. **Reshape Queries/Keys/Values:** The feature maps are reshaped to $(B, H \times W, C)$ for queries and $(B \times H \times W, K^2, C)$ for keys/values to prepare for localized attention computation.
5. **Multi-Head Attention:** The reshaped queries, keys, and values are processed through a multi-head attention module, computing attention scores within each neighbourhood independently using Equation 8:

$$Attention(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (8)$$

where d_k is the key dimension.

6. **Reshape to Original Dimensions:** The output is reshaped back to (B, H, W, C) and passed to subsequent network layers.

3.5. Proposed Lightweight Neural Network with NA

Table 6. Architecture of the proposed lightweight neural network with NA

Layer	Output	Parameters
Input	256 * 256 * 3	0
Seperable Conv2D	256 * 256 * 32	155
Batch Normalization	256 * 256 * 32	128
Max Pooling2D	128 * 128 * 32	0
Seperable Conv2D	128 * 128 * 64	2400
Seperable Conv2D	128 * 128 * 64	4736
Batch Normalization	128 * 128 * 64	256
Max Pooling2D	64 * 64 * 64	0
Seperable Conv2D	64 * 64 * 128	8896
Seperable Conv2D	64 * 64 * 128	17664
Batch Normalization	64 * 64 * 128	512
Max Pooling2D	32 * 32 * 128	0
Seperable Conv2D	32 * 32 * 256	34176
Seperable Conv2D	32 * 32 * 256	68096
Batch Normalization	32 * 32 * 256	1024
Seperable Conv2D	32 * 32 * 512	133888
Seperable Conv2D	32 * 32 * 512	267264
Batch Normalization	32 * 32 * 512	2048
Neighbourhood Attention	32 * 32 * 512	263040
Global Average Pooling	512	0
Dense	1568	804384
Drop Out	1568	0
Dense	38 (PlantVillage dataset) 100 (customised dataset)	59622 156,900
Total Trainable Parameters	PlantVillage dataset Customised dataset	1,668,289 1,765,567

The proposed model processes an input image of shape (256,256,3), representing a 256*256 pixel RGB image. It undergoes a series of transformations through several depth-wise separable convolutional and pooling layers designed to extract relevant features effectively as indicated in Table 6.

1. **Initial Convolutional Layers:** The architecture begins with *SeparableConv2D* layers, which reduce computational complexity compared to standard convolutions by factorizing them into depth-wise and pointwise convolutions. The first block applies *SeparableConv2D* (32, 3×3) followed by *Batch Normalization* and *MaxPooling2D* (2×2) to downsample the feature maps while preserving important local features.
2. **Progressive Feature Expansion:** Subsequent convolutional blocks increase the number of filters (64, 128, 256, 512) to progressively capture higher-level and more abstract features. Each block uses two stacked *SeparableConv2D* layers, followed by *Batch Normalization* to stabilize training, and *MaxPooling2D* (2×2) for spatial reduction.
3. **Neighbourhood Attention Module:** Instead of applying global SA, the architecture employs a *Neighbourhood Attention* block, parameterized with num_heads=4, key_dimensions=32, and neighbourhood_size=3. This mechanism focuses attention locally within a fixed-size window, improving computational efficiency and preserving local spatial relationships in the feature maps.
4. **Global Feature Aggregation:** After attention processing, *GlobalAveragePooling2D* summarizes the feature maps to form a single vector representation. It ensures invariance to spatial translations while reducing the number of trainable parameters.

5. **Fully Connected Layers for Classification:** The final dense layers process the aggregated features for classification. These layers integrate the learned spatial and contextual representations to produce the final output probabilities.

Advantages of the Proposed Design:

- *Efficiency:* Use of *SeparableConv2D* layers and local attention reduces computational cost.
- *Context-awareness:* NA captures relevant local dependencies while avoiding the high complexity of full SA.

4. Results

This section discusses the experimental design and evaluates the performance of the proposed lightweight neural network model using two datasets: the Plant Village dataset, and 100 category plant disease dataset. Various metrics are used including accuracy, loss, classification reports and confusion matrices to evaluate the performance of the model, as well as Grad-CAM analysis results, to show how the NA mechanism contributes to the model.

4.1. Experimental Setup

The proposed model was trained on Kaggle's GPU environment, which runs an NVIDIA Tesla P100 GPU (16 GB), and was implemented using the Keras API with a TensorFlow backend. The Plant Village dataset and the suggested 100-category dataset were used to train the model for 100 epochs, as Table 7. indicates. We employed sparse categorical cross-entropy as the loss function and an Adam optimizer with a learning rate of 0.0001.

Table 7. Hyperparameters for the proposed network

	Input image size	Optimizer	Split Ratio	Batch Size	Epochs	Dropout	Learning Rate	Loss Function
Plant Village	256 * 256 * 3	Adam	70:15:15	32	100 (35-epochs-early stopping)	0.5	0.0001	Sparse categorical
100-category dataset	256 * 256 * 3	Adam	70:15:15	32	100 (37-epochs-early stopping)	0.5	0.0001	Sparse categorical

4.2. Performance on Plant Village Dataset

Training and Validation Analysis - The training and validation results shown in Fig. 6. show a successful learning experience over 35 epochs. In the first five epochs, training accuracy increased rapidly from roughly 35% to over 90%. it reached a plateau at just under 98% by epoch ten. The trajectory of validation accuracy was similar. The training and validation accuracy scores show no misalignment or notable deviations, with the exception of small differences in epochs 9 and 15. Nevertheless, both curves are nearly equalizing at 98% by the end of training, offering compelling proof of generalization to fresh data.

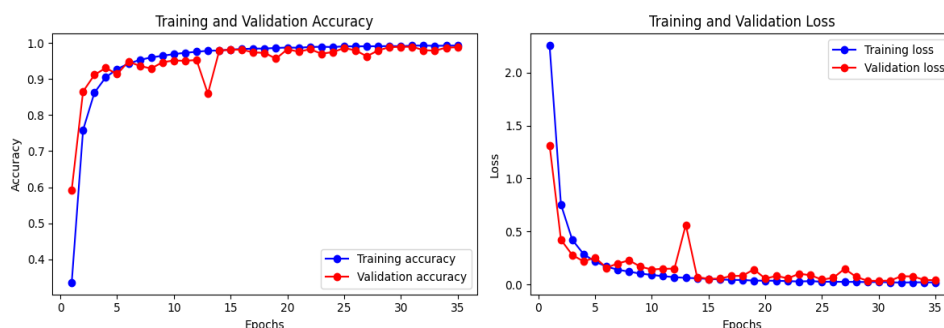


Fig. 6. Training and loss curves for PlantVillage dataset

In addition to validation loss, training loss rapidly decreased in the first few epochs from about 2.0 to less than 0.5 before plateauing below 0.2. Overall, the model was stable and maintained low error for the rest of training, although the slightly elevated validation loss observed in epoch 15 suggests brief overfitting.

Confusion Matrix and Classification Report - The confusion matrix outlined in Fig. 7. provides further assurance of the model's high performance for accurate classification across all 38 categories, characterized by strong diagonal dominance and quite a few classes, such as Classes 10, 16, 18, 23, and 28, resulting in nearly perfect accuracy. While there are occasional minor misclassifications (one example being Class 7 which at times is predicted as Class 38, and a small amount of confusion with Class 27), these misclassifications are very small in context with the total samples. The classification report (Appendix A) corroborates as most of the categories either have precision, recall, or F1-scores nearly

equal to 1.0. Perfect F1-scores are achieved for categories like Apple Black rot, Apple healthy, Corn Common rust, and Grape healthy. A few classes, in particular Corn Cercospora leaf spot and Tomato Target Spot that have similar symptoms that led to a slight decrease, from 0.98 to 0.95 to 0.97 range F1-score, but still resulting in a F1 score of around 0.96 to 0.97. Additionally, macro and weighted averages maximums of 0.99 solidify strong generalization, showing strong ability to classify a wide variety of plant diseases with minimal error.

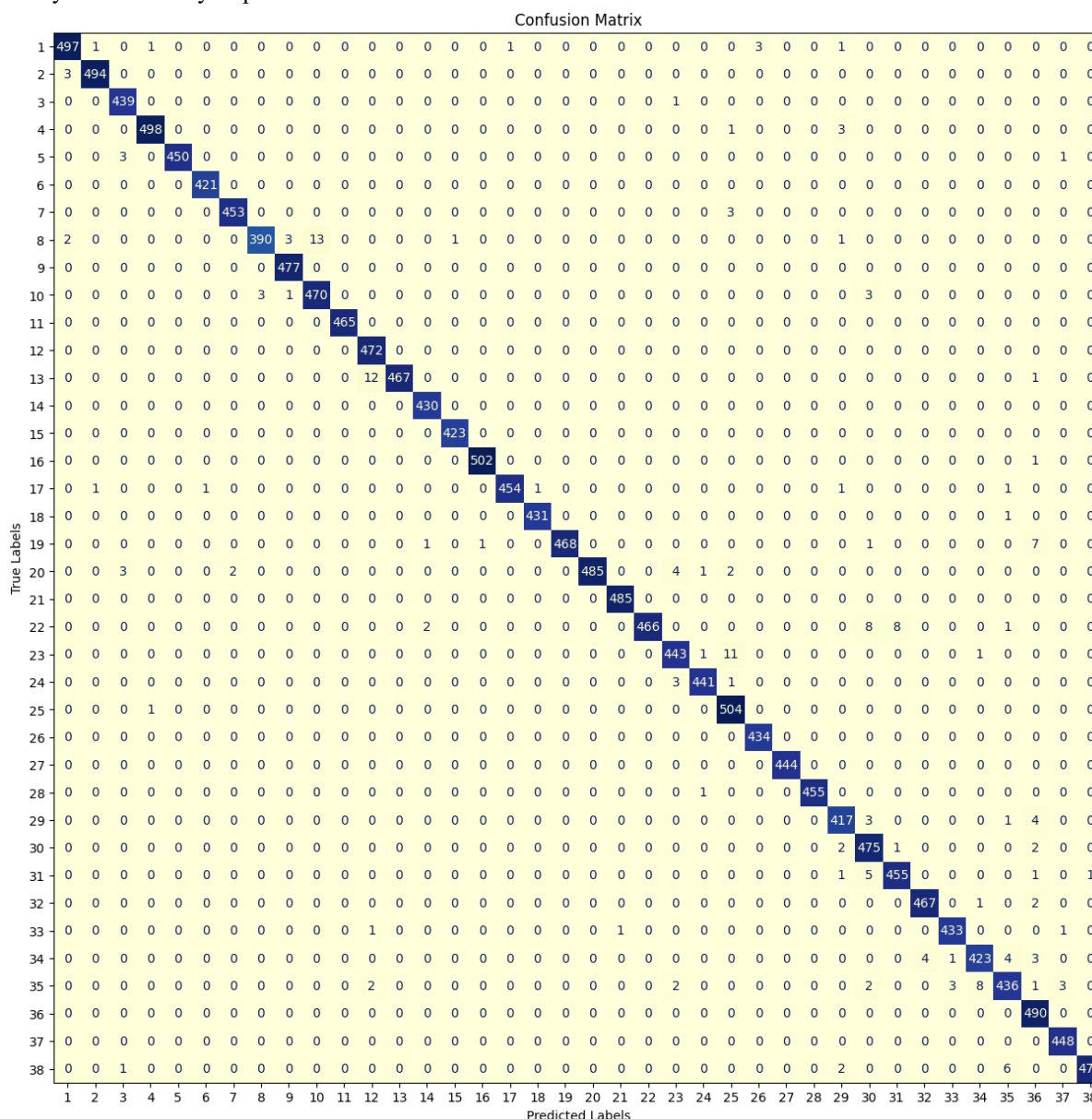


Fig. 7. Confusion Matrix for PlantVillage dataset

The classification report (Appendix A) confirms the proposed model achieves 99% overall accuracy and that almost all of the evaluated 38 categories achieved precision, recall, and F1-scores around or exactly at 1.0 (while total accuracy includes multiple categories achieving perfect F1-scores) except a few visually similar diseases which experienced slight reductions, and all average 0.99 confirms strong and consistent generalization across all plant disease categories.

4.3. Performance analysis on 100-Category Plant Disease Dataset

Training and Validation Analysis - The graphs for training and validation results presented in Fig. 8. indicate a steady increase in performance, with both accuracy metrics increasing in the early epochs and converging near 93–94% validation accuracy by the end of training, suggesting that the model can learn effectively without severe overfitting. The training and validation loss declined sharply in the initial epochs, with the validation loss remaining slightly higher than the training loss after the early epochs, and afterwards, both sets behaved consistently until the end of training. This confirms a stable and balanced training process. In addition, the results presented in the classification report show similarly positive performance with an overall accuracy of 93.3% across 100 classes. The macro averages for precision,

recall, and F1-score are 0.902, 0.894, and 0.896, respectively, while the weighted averages for precision, recall, and F1-score are 0.934, 0.933, and 0.933, respectively. Several of the 100 classes achieved perfect scores of 1.000, for example, classes 1, 20, 23, 24, 45, 46, 47, 52, 57, 58, 59, and 65. Most of the classes scored well, although a few, for example, class 8 with precision 0.554, recall 0.562, and F1-score 0.558, as well as classes 12, 13, 14, 21, and 97, did not perform as well in these evaluations — this performance can be attributed to higher inter-class similarity, limited test samples in some categories, and visual overlap among disease symptoms. Nevertheless, these results further confirmed that the model generalized well with the larger dataset. Class-specific results can be found in Appendix B. The confusion matrix has not been included here, as it is not visually accessible to summarize 100 separate classes.

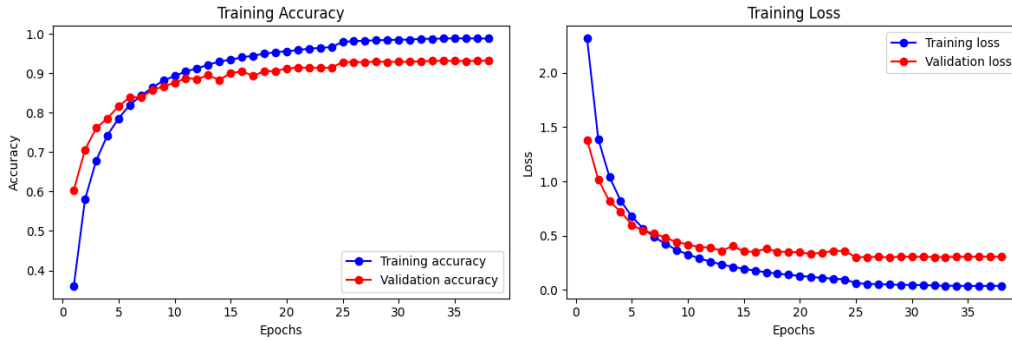


Fig. 8. Training and loss curves for 100 category plant disease dataset

4.4. Ablation Studies

Ablation studies were carried out to demonstrate the effectiveness of depth-wise separable convolutions and NA in the proposed architecture. As shown in Table 8, the standard convolutional baseline achieves high accuracy on both datasets ($98.89\% \pm 0.18\%$ on Plant Village and $95.66\% \pm 0.25\%$ on the 100-category dataset) but at the expense of a very large model size (21.25 MB) and parameter count (55.7 million), making it impractical for deployment. Substituting standard convolutions with depth-wise separable convolutions drastically reduces the parameters to 1.4 million and the model size to 5.36 MB, with only a small reduction in accuracy ($96.54\% \pm 0.30\%$ on Plant Village and $90.36\% \pm 0.30\%$ on the 100-category dataset). When NA is incorporated, the accuracy improves further, reaching $98.97\% \pm 0.33\%$ on Plant Village and $93.36\% \pm 0.28\%$ on the 100-category dataset, while still keeping the model lightweight (1.7 million parameters, 6.74 MB). These ablation models were trained and evaluated on both datasets for five independent runs to ensure a fair and consistent comparison, demonstrating that NA provides additional performance gains with only a minimal increase in parameters.

Table 8. Ablation study results on the 100-category plant disease dataset, comparing standard convolutions, depthwise separable convolutions, and NA in terms of model size, parameters, and test accuracy.

Model Variant	Model size (MB)	Parameters (Millions)	Plant Village Accuracy	100- category Accuracy
Proposed model (Standard Convolutions)	21.25	55.7	$98.89\% \pm 0.18\%$	$95.66\% \pm 0.25\%$
Proposed model (Depth-wise Separable Convolutions without NA)	5.36	1.4	$96.54\% \pm 0.30\%$	$90.36\% \pm 0.30\%$
Proposed Model (Depth-wise Separable Convolutions with NA)	6.74	1.7	$98.97\% \pm 0.33\%$	$93.36\% \pm 0.28\%$

4.5. Computational Complexity And Scalability Analysis

The computational complexity of the proposed model was analyzed to clarify its scalability with increasing input resolution. The proposed CNN consists mainly of depth-wise separable convolutional layers and a localized Neighborhood Attention (NA) block. Therefore, the complexity analysis considers both the convolutional component and the localized attention component.

Let the input feature map be represented as $X \in \mathbb{R}^{H \times W \times C}$ where H , W , and C denote the height, width, and number of channels, respectively. The number of spatial tokens is given by $N = H \times W$. In conventional global self-attention, each token attends to every other token in the feature map. Therefore, the attention map contains $N \times N$ pairwise interactions, resulting in a computational and memory complexity of $O(N^2)$.

In contrast, the localized NA block used in the proposed model restricts attention to a fixed local neighborhood of size $k \times k$ around each spatial position. Thus, each token interacts with only k^2 neighboring tokens instead of all N tokens. The complexity of the localized attention operation can therefore be expressed as: $O(Nk^2)$.

Since the neighborhood size k is fixed and independent of the input image resolution, k^2 can be treated as a constant. Hence, the localized NA operation scales approximately linearly with the number of spatial tokens: $O(Nk^2) = O(N)$.

The convolutional component also supports efficient scaling through depth-wise separable convolution which has a cost $O(H*W*C_{in}(k^2+C_{out}))$. For fixed kernel size and channel dimensions, this cost grows proportionally with $H*W$. Thus, both the localized NA block and the depth-wise separable convolutional layers support approximately linear scaling with the number of spatial positions.

To empirically verify the scalability of the proposed model, the trained model was evaluated on the test dataset using different input resolutions, namely 64*64, 128*128, 256*256, 384*384, 512*512. For each input size, the number of input pixels, input memory requirement, GFLOPs, and average inference time per image were measured. The results are presented in Table 9.

Table 9. Computational scalability analysis of the proposed CNN on the test dataset at different input resolutions

Input Size	Input Pixels	Input Memory (MB)	GFLOPs	Avg. Inference Time/Image (s)
64*64	4,096	4.6875	0.0962	0.0151
128*128	16,384	18.7500	0.3792	0.040
256*256	65,536	75.0000	1.5110	0.1033
384*384	147,456	168.7500	3.3989	0.2877
512*512	262,144	300.0000	6.0393	0.3294

As shown in Table 9, GFLOPs, input tensor memory, and average inference time increase with input resolution. When the resolution increases from 64×64 to 128×128, the number of pixels increases by 4×, while GFLOPs increase by approximately 3.94×. Similarly, from 128×128 to 256×256, GFLOPs increase by approximately 3.98×, and from 256×256 to 512×512, the increase is approximately 4×. This indicates approximately linear computational scaling with respect to the number of input pixels. The inference time also increases monotonically from 0.0151 s/image to 0.3294 s/image, confirming the expected runtime increase at higher resolutions.

4.6. Grad-CAM Analysis and Occlusion Sensitivity Test

In Fig. 9., the upper row shows the original photographs of the leaves of plants that display recognizable symptoms of disease, while the middle and lower rows illustrate Grad-CAM into the images without and with NA blocks respectively. The middle row identifies what parts of the images, the model without NA was attending to. As shown, the model focused on some areas deserving of attention, although the heatmaps lack precision and simply spread attention across irrelevant parts of the images.

The lower row compares the attentional focus from the version of model without NA blocks by showing the heatmaps produced by using NA blocks. In this row, the model is more precise, focusing on areas actually affected by disease. More careful attention focus means the model can better identify plant diseases.

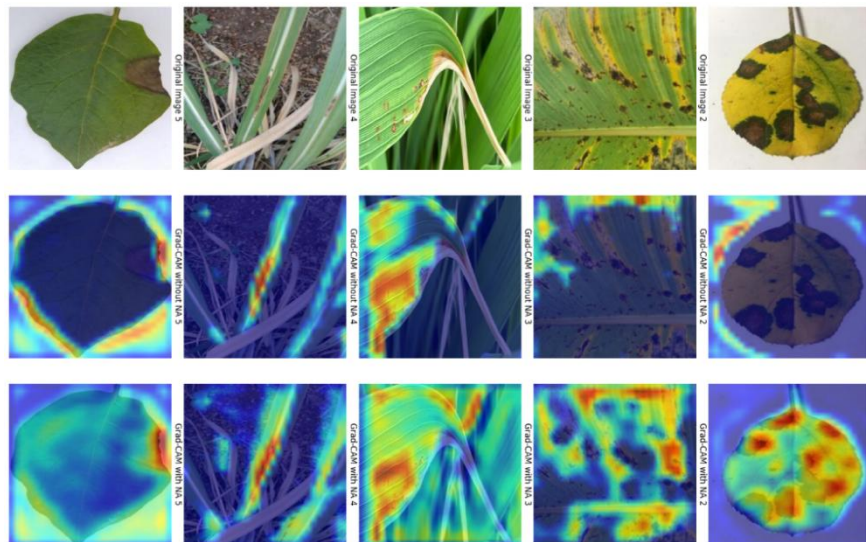


Fig. 9. Grad-CAM visualizations comparing feature localization with and without the proposed NA block. The NA-enhanced model consistently focuses on precise disease-affected regions, whereas the model without NA produces diffused and less discriminative attention maps.

To further validate the Grad-CAM observations, prediction-guided occlusion sensitivity analysis was performed on correctly classified test images. For each image, Grad-CAM was generated with respect to the model's predicted class. The highly activated Grad-CAM regions were then occluded, and the confidence score for the same predicted class was recomputed. The confidence drop was calculated as: $\Delta C = C_{original} - C_{occluded}$. A larger confidence drop indicates that the highlighted region contains important visual evidence used by the model for classification.

Out of 6,488 test images, 6,052 were correctly classified and used for occlusion sensitivity analysis. The mean confidence decreased from 0.9808 to 0.4393 after occlusion, resulting in a mean confidence drop of 0.5416 and a median confidence drop of 0.6339. This substantial reduction indicates that the Grad-CAM-highlighted regions strongly influence the model's prediction and that the proposed CNN primarily relies on disease-relevant visual regions for correct classification. Representative examples are shown in Fig. 10.

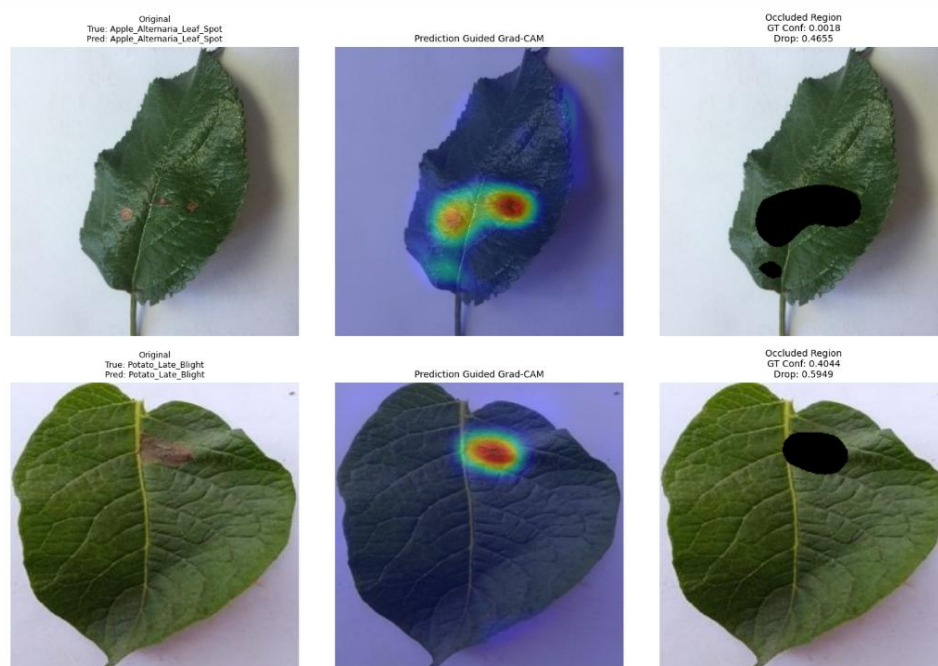


Fig. 10. Grad-CAM and occlusion sensitivity results -The figure shows original leaf images, prediction guided Grad-CAM heatmaps, and occluded images. The confidence drops after occluding the highlighted disease regions indicates that the model focuses on relevant symptomatic areas for classification.

4.7. Comparison of proposed model with state of art

To assess the performance of the presented CNN architecture, a comparison was made with several top models in deep learning, including MobileNetV1, MobileNetV2, DenseNet121, InceptionV3, MobileViT-XXS, and EfficientViT-M0. All the models were trained and tested using the same experimental settings on the 100-Category Plant Disease Dataset as well as the Plant Village Dataset. To assess statistical reliability, the proposed model was evaluated over five independent runs using different random seeds. On the 100-category dataset, the model achieved a mean test accuracy of $93.36\% \pm 0.28\%$, with a 95% confidence interval of $93.02\% - 93.71\%$. The macro precision, macro recall, and macro F1-score were $90.30\% \pm 0.68\%$, $90.04\% \pm 0.66\%$, and $89.99\% \pm 0.69\%$, with 95% confidence intervals of $89.46\% - 91.14\%$, $89.22\% - 90.86\%$, and $89.14\% - 90.84\%$, respectively. The narrow confidence intervals indicate stable and reproducible performance across repeated runs.

As shown in Table 10, the proposed model achieved the best overall performance on the 100-category dataset among the compared models. This performance is significant given the difficulty of the dataset, which contains a large number of visually similar disease classes. The proposed model maintained a compact architecture with 1.76 million parameters and a storage size of 6.74 MB, which is smaller than MobileNetV1 (12.87 MB), MobileNetV2 (9.29 MB), DenseNet121 (27.40 MB), and InceptionV3 (84.22 MB). While MobileViT-XXS was smaller at 4.60 MB and EfficientViT-M0 had lower computational complexity with 0.16 GFLOPs, their accuracies on the 100-category dataset were lower, at 91.80% and 92.50%, respectively.

Table 10. Comparison of performance of proposed model and state of art CNN on the 100-category plant disease dataset and PlantVillage dataset

Model	Parameters (Millions)	Model Size (MB)	GFLOPs	100 category dataset		PlantVillage dataset	
				Test accuracy (%)	F1 score (%)	Test accuracy (%)	F1 score (%)
MobileNetV1	3.37	12.87	1.14	87.65	87.55	98	97.97
MobileNetV2	2.43	9.29	0.60	89.24	89.19	96.11	96.07

DenseNet121	7.18	27.40	5.70	90.25	90.24	93.39	93.33
InceptionV3	22.07	84.22	4.00	91.25	91.24	96.43	96.41
MobileViT-XXS	1.4	4.60	0.40	91.80	91.75	95.45	95.42
EfficientViT-M0	2.4	7.23	0.16	92.50	92.42	95.69	95.53
Proposed CNN	1.76	6.74	1.51	93.36 $\pm$ 0.28	93.26 $\pm$ 0.28	98.97 $\pm$ 0.33	98.88 $\pm$ 0.33

The proposed model also performed strongly on the PlantVillage dataset, achieving 98.97% $\pm$ 0.33% test accuracy and 98.88% $\pm$ 0.33% F1-score. The results suggest that the proposed CNN presents a competitive trade-off between performance, model size, and computational efficiency across datasets.

4.8. Comparison of the proposed model with existing literature

This section of the paper compares CNN model performance to existing state-of-the-art lightweight architectures published in the literature for plant disease detection in Table 11.

Table 11. Comparison of Lightweight Models for Plant Disease Detection

No.	Ref	Model Used	Plant Species	Dataset	Parameters & Model Size	Accuracy (%)
1	[43]	MobileViT DAP	Rice	Self-captured; 11k images; 6 classes	0.75 million params	99.61
2	[44]	MobilePlantViT	14 plants	Plant Village	0.69 million params	99.57
3	[53]	Optimized ViT	14 plants	Plant Village	13 million params	99.77
4	[54]	RTR Lite MobileNet	—	Plant Diseases Dataset	1.05 million params; 4.01 MB	99.92
5	[55]	CropNet	Wheat	Self-captured; 5k images; 5 classes	6 million params; 51 MB	99.80
6	[56]	MobileNetV3 Small	Cassava	Cassava Image Dataset; 8k images; 6 classes	1.67 million params	98.95
7	[57]	BEiT	14 plants	Plant Village	12 million params	97.33
8	[58]	LDAMNet	Tomato	Plant Village; 20k images; 10 classes	0.91 million params	98.71
9		Proposed Architecture	16 plants	Customized 100-category dataset	1.7 million params; 6.74 MB	93.36 $\pm$ 0.28

The lightweight CNN model proposed in this study achieves 93.36% $\pm$ 0.28% accuracy on a customized 100-category dataset consisting of 16 plant species. Although several existing models report higher accuracies, many of them were evaluated on single-crop datasets, fewer disease classes, or controlled datasets such as PlantVillage. For example, MobileViT-DAP was tested only on rice with six classes, CropNet on wheat with five classes, and LDAMNet on tomato diseases. Similarly, transformer-based models such as Optimized ViT and BEiT report high accuracy but require larger parameter counts, which may limit deployment on low-resource devices.

In contrast, the proposed architecture is trained and evaluated on a broader multi-species dataset with 100 categories, making the classification task more challenging due to increased inter-class similarity and dataset diversity. With 1.76 million parameters and a model size of 6.74 MB, the proposed model provides a practical balance between accuracy, compactness, and scalability. Therefore, while some literature models achieve higher accuracy on smaller or more controlled datasets, the proposed model demonstrates competitive performance for multi-crop plant disease detection and remains suitable for mobile or embedded agricultural applications.

4.9. Limitations and Failure Case Analysis

- Variation in Class-wise Performance** - Although the proposed lightweight CNN with neighbourhood attention performed well overall, some limitations were observed. The proposed model achieved overall accuracy of 93.3% and weighted F1 score of 93.3% on the 100-category dataset. However, class-wise performance was not uniform across all categories. The lower macro F1-score of 89.6% as opposed to the weighted F1-score suggests that some minority and visually similar classes were harder to classify.
- Failure Cases in Banana Nutrient-Deficiency Classes** - The main failure cases were observed in banana nutrient-deficiency categories. Classes such as Banana Manganese Deficient, Banana Iron Deficient, Banana Magnesium Deficient, Banana Zinc Deficient, and Banana Calcium Deficient obtained lower F1-scores compared with most other classes. This may be due to the high visual similarity among nutrient-deficiency symptoms. Symptoms such as chlorosis, yellowing, marginal drying, necrotic patches, and irregular discoloration often overlap across different deficiencies, making fine-grained classification challenging.
- Dataset Construction Limitations** - Another limitation relates to the construction of the dataset. The proposed 100-class dataset is created by merging images from different open sources. While duplicate images were eliminated with perceptual hashing and images were manually checked for label consistency, there may still be differences in image resolution, background, illumination, disease severity and acquisition conditions. Also, some classes originally had fewer natural images and required more augmentation to balance the training data. Although augmentation increases the number of training samples, it may not be able to fully substitute the diversity of field images that are captured naturally.

4. **Architectural Limitations** - The proposed architecture was designed to achieve a balance between classification accuracy and computational efficiency. The use of depth-wise separable convolutions significantly reduces the number of parameters and model size compared with standard convolutions, making the model more suitable for lightweight deployment. However, this efficiency is achieved by reducing full cross-channel convolutional interactions, which may affect feature representation in highly complex cases. Future work may investigate whether limited additional channel interaction can further improve representation without substantially increasing model size or computational cost. Similarly, the localized neighbourhood attention block improves the model's ability to focus on disease-relevant regions while avoiding the high cost of global self-attention. Nevertheless, because the attention is restricted to local neighbourhoods, the model may face difficulty when symptoms are extremely small, weakly visible, and widely scattered. Therefore, future work may explore multi-scale neighbourhood attention or hybrid local-global attention mechanisms to further improve robustness while preserving the lightweight nature of the model.
5. **Scalability and Deployment Limitations** - The scalability analysis using different input sizes showed that the proposed model can operate across varying resolutions. Increasing the input resolution also increases FLOPs, memory usage, and inference latency. Therefore, although the model is suitable for lightweight plant disease classification, practical deployment on mobile or edge devices requires hardware-specific evaluation in terms of real-time inference speed, memory consumption, and energy usage.
6. **Future Improvement Directions** - Overall, these results indicate that the proposed model is effective for large-scale 100-class plant disease classification, but further improvement is needed for visually similar nutrient-deficiency classes, underrepresented categories, and real-field deployment conditions. Future work will focus on increasing naturally collected field images for weak classes, improving fine-grained feature discrimination, performing confusion-matrix-based error analysis, and validating the model on mobile or edge platforms.

5. Conclusion

This manuscript deals with several relevant issues regarding the detection of plant diseases, such as the large model size and number of parameters associated with traditional CNNs, the high computational cost related to self-attention mechanisms, and the drawbacks associated with the PlantVillage dataset. First, a simple and efficient lightweight neural network architecture for plant disease detection is built using depth-wise separable convolutions combined with neighborhood attention (NA) within a 9-layer CNN. The proposed NA reduces the computational burden of global self-attention by attending only to local regions, thereby enabling efficient extraction of disease-relevant local patterns. By employing depth-wise separable convolutions alongside the NA block, the proposed model achieves both efficiency and accuracy by focusing computations on the most relevant local disease features. Second, to overcome the limitations of the PlantVillage dataset, which has a uniform background and low variability, we developed a larger and more varied dataset by collecting images from multiple open-source datasets and applying augmentation only to the training set. The new dataset includes plant leaf images across 16 plant species, organized into 100 plant disease and healthy classes. The lightweight model was fully trained using both the PlantVillage dataset and the newly created 100-category plant disease dataset. It achieved a high accuracy of $98.97\% \pm 0.33\%$ with an F1-score of $98.88\% \pm 0.33\%$ on the PlantVillage dataset and $93.36\% \pm 0.28\%$ accuracy with an F1-score of $93.26\% \pm 0.28\%$ on the newly created 100-category dataset. The results of this framework show competitive performance compared with advanced models such as MobileNetV1, MobileNetV2, DenseNet121, InceptionV3, MobileViT-XXS, and EfficientViT-M0 under the same experimental settings. Depth-wise separable convolution helped reduce the parameter count to 1.76 million and the model size to just 6.74 MB, while neighborhood attention improved local feature representation by focusing on disease-relevant regions. The proposed framework shows strong potential for practical plant disease diagnosis, particularly in applications requiring lightweight and efficient deep learning models. Future work may explore disease severity estimation and integration into mobile or IoT-based field-deployable systems. Furthermore, the proposed model can be further optimized by applying quantization techniques, enabling faster inference and lower memory usage on edge devices.

All the Declarations and Statements

Author Contribution Statement

Rithambara Rajput - Conceptualization, methodology, dataset collection and curation, data preprocessing, duplicate removal, augmentation, model design, implementation, training, validation, testing, ablation study, computational analysis, Grad-CAM and occlusion sensitivity analysis, result interpretation, manuscript drafting, revision, and formatting.

Suneeta Budihal - Supervision, research guidance, review of the methodology, manuscript review, technical suggestions, and overall project monitoring.

Saroja Siddamal - Supervision, academic guidance, manuscript review, result interpretation support, and critical

feedback for improving the manuscript.

Dattaprasad Torse - Supervision, technical guidance, manuscript review, validation of the research direction, and critical comments for manuscript refinement.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflict of interest

Funding Declaration

This research work was supported by KLE Technological University, Hubballi, India under the Ph.D. Fellowship Program.

Data Availability Statement

The newly curated 100-category Plant Disease Dataset used in this study is publicly available on Kaggle.

Augmented Train Dataset: <https://www.kaggle.com/datasets/rithambararajput/augmented-train>

Raw Dataset: <https://www.kaggle.com/datasets/rithambararajput/100-class-split-raw-dataset>

The Plant Village dataset, used as a benchmark for comparative evaluation, is also publicly accessible at: <https://www.kaggle.com/datasets/vipooool/new-plant-diseases-dataset>.

Ethical Declarations

This study does not involve human participants or animals. Therefore, ethical approval was not required.

Acknowledgements

The authors express their gratitude to KLE Technological University, Hubballi, India, and KLE Technological University's Dr. M. S. Sheshgiri Campus, Belagavi, for their academic support and research facilities during this work.

Declaration of Generative AI in Scholarly Writing

During the preparation of this manuscript, the authors used AI-assisted tools to improve language clarity, grammar, formatting, and organization of the manuscript. The authors reviewed, edited, and verified all AI-assisted content and take full responsibility for the final content of the manuscript, including the accuracy of data, results, analysis, and conclusions.

Abbreviations

The following abbreviations are used in this manuscript:

AI – Artificial Intelligence

DL – Deep Learning

CNN – Convolutional Neural Network

ViT – Vision Transformer

SA – Self-Attention

NA – Neighbourhood Attention

NAT – Neighbourhood Attention Transformer

RGB – Red, Green, Blue

GPU – Graphics Processing Unit

MB – Megabyte

GB – Gigabyte

FLOPs – Floating Point Operations

GFLOPs – Giga Floating Point Operations

Grad-CAM – Gradient-weighted Class Activation Mapping

IoT – Internet of Things

API – Application Programming Interface

ECA – Efficient Channel Attention

CBAM – Convolutional Block Attention Module

CA – Channel Attention

Appendix A\B\C..., with appendix tile

Appendix A: Classification Report for Plant Village Dataset

The model exhibits high classification performance across the 38 categories with an overall accuracy rate of 98.87%.

For most categories, precision, recall, and F1-scores were all close values, indicating strong predictive ability of the model overall. Specifically, there were perfect scores for some classes, including categories 0, 1, 4, 10, 12, and 26 providing high levels of discrimination reliability. Some classes were relatively low on recall values which include the range of category 7 (F1 = 0.9714) and category 21 (F1 = 0.9800) levels. Still, the macro-averaged precision and F1 were 0.9890 and 0.9888 respectively providing near-similar weighted averages at acceptable and well-balanced levels of performance throughout the data set overall. Overall, the model performed well with few opportunities for improvement across a small range of challenging categories.

Label	precision	recall	f1-score	support
0	0.9900	0.9861	0.9881	504.0000
1	0.9960	0.9940	0.9950	497.0000
2	0.9843	0.9977	0.9910	440.0000
3	0.9960	0.9920	0.9940	502.0000
4	1.0000	0.9912	0.9956	454.0000
5	0.9976	1.0000	0.9988	421.0000
6	0.9956	0.9934	0.9945	456.0000
7	0.9924	0.9512	0.9714	410.0000
8	0.9917	1.0000	0.9958	477.0000
9	0.9731	0.9853	0.9792	477.0000
10	1.0000	1.0000	1.0000	465.0000
11	0.9692	1.0000	0.9844	472.0000
12	1.0000	0.9729	0.9863	480.0000
13	0.9931	1.0000	0.9965	430.0000
14	0.9976	1.0000	0.9988	423.0000
15	0.9980	0.9980	0.9980	503.0000
16	0.9978	0.9891	0.9934	459.0000
17	0.9977	0.9977	0.9977	432.0000
18	1.0000	0.9791	0.9894	478.0000
19	1.0000	0.9759	0.9878	497.0000
20	0.9979	1.0000	0.9990	485.0000
21	1.0000	0.9608	0.9800	485.0000
22	0.9779	0.9715	0.9747	456.0000
23	0.9932	0.9910	0.9921	445.0000
24	0.9655	0.9980	0.9815	505.0000
25	0.9931	1.0000	0.9966	434.0000
26	1.0000	1.0000	1.0000	444.0000
27	1.0000	0.9978	0.9989	456.0000
28	0.9743	0.9812	0.9777	425.0000
29	0.9557	0.9896	0.9724	480.0000
30	0.9806	0.9827	0.9817	463.0000
31	0.9915	0.9936	0.9926	470.0000
32	0.9908	0.9931	0.9920	436.0000
33	0.9769	0.9724	0.9747	435.0000
34	0.9689	0.9540	0.9614	457.0000
35	0.9570	1.0000	0.9780	490.0000
36	0.9890	1.0000	0.9945	448.0000
37	0.9979	0.9813	0.9895	481.0000
accuracy	0.9887			
macro avg	0.9890	0.9887	0.9888	17572.0000
weighted avg	0.9889	0.9887	0.9887	17572.0000

Table A: Classification Report for Plant Village Dataset

Appendix B: Classification Report for 100-Category plant disease Dataset

The model exhibits strong classification performance across the 100 categories, achieving an overall accuracy of 93.3%. The majority of the classes obtained high precision, recall, and F1-scores. Some classes, such as label 1: Apple Brown Spot, label 20: Banana Yellow Sigatoka Disease, label 23: Coffee Miner, label 24: Coffee Phoma, label 45: Grapes Healthy, label 46: Grapes Leaf Blight, label 47: Guava Canker, label 52: Mango Anthracnose, label 57: Mango Powdery Mildew, label 58: Mango Sooty Mould, label 59: Mango Weevil, and label 65: Rice Bacterial Blight, achieved perfect precision, recall, and F1-scores. This indicates that the model classified these categories without false positives or false negatives in the test set.

However, comparatively lower performance was observed for some categories, namely label 8: Banana Calcium Deficient, label 12: Banana Iron Deficient, label 13: Banana Magnesium Deficient, label 14: Banana Manganese

Deficient, label 19: Banana Sulphur Deficient, label 21: Banana Zinc Deficient, label 67: Rice Leaf Smut, label 77: Tea Anthracnose, and label 97: Wheat Loose Smut. For example, Banana Calcium Deficient obtained a precision of 0.554 and an F1-score of 0.558, Banana Iron Deficient obtained an F1-score of 0.222, Banana Magnesium Deficient obtained an F1-score of 0.370, Banana Zinc Deficient obtained an F1-score of 0.404, and Wheat Loose Smut obtained an F1-score of 0.708. These results indicate that certain visually similar or low-sample categories may require additional representative samples. Overall, the proposed architecture demonstrates strong classification capability, while a small number of categories still require improvement.

Table B: Classification Report for 100-Category Plant Disease Dataset

Label	precision	recall	f1-score	support
0	0.976	0.976	0.976	42
1	1.000	1.000	1.000	33
2	0.964	0.898	0.930	59
3	0.969	1.000	0.984	62
4	0.980	0.962	0.971	52
5	1.000	0.727	0.842	11
6	0.667	0.667	0.667	15
7	1.000	0.875	0.933	8
8	0.554	0.562	0.558	64
9	1.000	0.920	0.958	25
10	1.000	0.923	0.960	13
11	0.833	0.769	0.800	13
12	0.214	0.231	0.222	13
13	0.323	0.435	0.370	23
14	0.000	0.000	0.000	4
15	1.000	0.778	0.875	9
16	0.778	1.000	0.875	7
17	0.839	1.000	0.912	26
18	0.688	0.710	0.698	31
19	0.647	0.600	0.623	110
20	1.000	1.000	1.000	4
21	0.426	0.383	0.404	60
22	0.915	0.900	0.908	60
23	1.000	1.000	1.000	54
24	1.000	1.000	1.000	58
25	0.846	0.917	0.880	60
26	0.921	0.933	0.927	150
27	0.979	0.933	0.956	150
28	0.865	0.895	0.880	86
29	0.980	0.987	0.983	150
30	0.982	0.982	0.982	56
31	0.930	0.930	0.930	57
32	0.883	0.946	0.914	56
33	0.948	0.982	0.965	56
34	0.943	0.893	0.917	56
35	0.911	0.927	0.919	55
36	0.788	0.867	0.825	30
37	0.852	0.767	0.807	30
38	0.840	0.700	0.764	30
39	0.867	0.867	0.867	30
40	0.848	0.933	0.889	30
41	0.765	0.867	0.812	30
42	1.000	0.900	0.947	30
43	0.972	0.972	0.972	71
44	0.986	1.000	0.993	72
45	1.000	1.000	1.000	64
46	1.000	1.000	1.000	65
47	1.000	1.000	1.000	25
48	0.938	0.938	0.938	16
49	0.991	1.000	0.996	116
50	0.962	1.000	0.980	25
51	0.963	1.000	0.981	26
52	1.000	1.000	1.000	72
53	0.987	1.000	0.993	74
54	1.000	0.946	0.972	37
55	0.973	1.000	0.986	73
56	1.000	0.986	0.993	74
57	1.000	1.000	1.000	74

58	1.000	1.000	1.000	75
59	1.000	1.000	1.000	75
60	0.970	0.962	0.966	133
61	0.963	0.981	0.972	106
62	0.980	0.973	0.977	150
63	0.960	0.992	0.975	120
64	0.980	0.973	0.977	150
65	1.000	1.000	1.000	6
66	1.000	0.833	0.909	6
67	0.750	0.500	0.600	6
68	0.985	0.970	0.977	66
69	0.870	0.940	0.904	50
70	0.931	0.931	0.931	58
71	0.972	0.972	0.972	72
72	0.966	0.982	0.974	57
73	0.949	0.949	0.949	78
74	0.970	0.956	0.963	68
75	0.987	0.987	0.987	75
76	1.000	0.882	0.938	17
77	0.909	0.667	0.769	15
78	0.824	0.933	0.875	15
79	0.850	1.000	0.919	17
80	0.824	0.933	0.875	15
81	1.000	0.917	0.957	12
82	0.957	1.000	0.978	22
83	0.905	0.864	0.884	22
84	0.951	0.900	0.925	150
85	0.946	0.940	0.943	150
86	0.986	0.960	0.973	150
87	0.957	0.893	0.924	150
88	0.925	0.980	0.951	150
89	0.986	0.960	0.973	150
90	0.887	0.975	0.929	121
91	0.890	0.913	0.901	150
92	0.980	0.980	0.980	150
93	0.967	0.980	0.974	150
94	0.980	0.980	0.980	150
95	0.948	0.955	0.952	134
96	0.973	0.947	0.959	150
97	0.852	0.605	0.708	38
98	0.800	0.952	0.870	42
99	0.985	0.993	0.989	135
accuracy	0.933			
macro avg	0.902	0.894	0.896	6488
weighted avg	0.934	0.933	0.933	6488

References

- [1] G. Bv, & U. D. G. (2019). A survey on plant disease prediction using machine learning and deep learning techniques. *International Journal of Artificial Intelligence*, 22(63), 0–18. DOI: 10.4114/intartif.vol22iss63pp0-19
- [2] K. Alibabaei et al. (2022). A review of the challenges of using deep learning algorithms to support decision-making in agricultural activities. *Remote Sensing*, 14(3), 1–43. DOI: 10.3390/rs14030638
- [3] C. Jackulin, & S. Murugavalli. (2022). A comprehensive review on detection of plant disease using machine learning and deep learning approaches. *Measurement: Sensors*, 24, 100441. DOI: 10.1016/j.measen.2022.100441
- [4] A. Krizhevsky, I. Sutskever, & G. E. Hinton. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. DOI: 10.1145/3065386
- [5] K. Simonyan. (2015). Very deep convolutional networks for large scale image recognition. *International Conference on Learning Representations (ICLR 2015)*, 1–14.
- [6] K. He, X. Zhang, S. Ren, & J. Sun. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. DOI: 10.1109/CVPR.2016.90
- [7] S. Wang, B. Z. Li, M. Khabsa, H. Fang, & H. Ma. (2020). Linformer: Self-attention with linear complexity. *arXiv preprint*. Available: <http://arxiv.org/abs/2006.04768>
- [8] S. Mehta, & M. Rastegari. (2022). MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer. *International Conference on Learning Representations (ICLR 2022)*.
- [9] Z. Liu, H. Mao, C. Y. Wu, C. Feichtenhofer, T. Darrell, & S. Xie. (2022). A ConvNet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11966–11976. DOI: 10.1109/CVPR52688.2022.01167
- [10] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, & Y. Yuan. (2023). EfficientViT: Memory efficient vision transformer with cascaded group attention. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14420–14430. DOI:

10.1109/CVPR52729.2023.01386

- [11] K. Wu et al. (2022). TinyViT: Fast pretraining distillation for small vision transformers. *Lecture Notes in Computer Science*, 13681, 68–85. DOI: 10.1007/978-3-031-19803-8_5
- [12] R. Kumar, A. Chug, A. P. Singh, & D. Singh. (2022). A systematic analysis of machine learning and deep learning based approaches for plant leaf disease classification: A review. *Journal of Sensors*, 2022. DOI: 10.1155/2022/3287561
- [13] H. Li, X. Yue, C. Zhao, & L. Meng. (2023). Lightweight deep neural network from scratch. *Journal Name Not Provided*, 18868–18886.
- [14] A. Hassani, S. Walton, J. Li, S. Li, & H. Shi. (2022). Neighborhood attention transformer. *arXiv preprint*. DOI: 10.48550/arXiv.2204.07143
- [15] L. Kaiser, F. Chollet, & A. N. Gomez. (2018). Depthwise separable convolutions for neural machine translation. *International Conference on Learning Representations (ICLR 2018)*, 1–10.
- [16] A. Hassani, & H. Shi. (2022). Dilated neighborhood attention transformer. *arXiv preprint*. DOI: 10.48550/arXiv.2209.15001
- [17] S. P. Mohanty, D. P. Hughes, & M. Salathé. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, 1–10. DOI: 10.3389/fpls.2016.01419
- [18] J. Amara, B. Bouaziz, & A. Algergawy. (2017). A deep learning-based approach for banana leaf diseases classification. *Lecture Notes in Informatics*, 79–88.
- [19] L. G. Nachtigall, R. M. Araujo, & G. R. Nachtigall. (2017). Classification of apple tree disorders using convolutional neural networks. *Proceedings of the 2016 IEEE 28th International Conference on Tools with Artificial Intelligence*, 472–476. DOI: 10.1109/ICTAI.2016.75
- [20] J. G. Ha et al. (2017). Deep convolutional neural network for classifying Fusarium wilt of radish from unmanned aerial vehicles. *Journal of Applied Remote Sensing*, 11(04). DOI: 10.1117/1.jrs.11.042621
- [21] D. Oppenheim, & G. Shani. (2017). Potato disease classification using convolution neural networks. *Advances in Animal Biosciences*, 8(2), 244–249. DOI: 10.1017/S2040470017001376
- [22] M. Brahimi, K. Boukhalfa, & A. Moussaoui. (2017). Deep learning for tomato diseases: Classification and symptoms visualization. *Applied Artificial Intelligence*, 31(4), 299–315. DOI: 10.1080/08839514.2017.1315516
- [23] K. P. Ferentinos. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311–318. DOI: 10.1016/j.compag.2018.01.009
- [24] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, & D. P. Hughes. (2017). Deep learning for image-based cassava disease detection. *Frontiers in Plant Science*, 8, 1–7. DOI: 10.3389/fpls.2017.01852
- [25] K. Zhang, Q. Wu, A. Liu, & X. Meng. (2018). Can deep learning identify tomato leaf disease? *Advances in Multimedia*, 2018. DOI: 10.1155/2018/6710865
- [26] M. Sibiya, & M. Sumbwanyambe. (2019). A computational procedure for the recognition and classification of maize leaf diseases out of healthy leaves using convolutional neural networks. *AgriEngineering*, 1(1), 119–131. DOI: 10.3390/agriengineering1010000
- [27] I. Z. Mukti, & D. Biswas. (2019). Transfer learning based plant diseases detection using ResNet50. *Proceedings of the 2019 4th International Conference on Electrical Information and Communication Technology*, 1–6. DOI: 10.1109/EICT48899.2019.9068805
- [28] S. T., R. Khilar, & M. Subaja Christo. (2021). A comparative analysis on plant pathology classification using deep learning architecture – ResNet and VGG19. *Materials Today: Proceedings*. DOI: 10.1016/j.matpr.2020.11.993
- [29] L. Mkonyi et al. (2020). Early identification of *Tuta absoluta* in tomato plants using deep learning. *Scientific African*, 10. DOI: 10.1016/j.sciaf.2020.e00590
- [30] E. Elfatimi, R. Eryigit, & L. Elfatimi. (2022). Beans leaf diseases classification using MobileNet models. *IEEE Access*, 10. DOI: 10.1109/ACCESS.2022.3142817
- [31] J. V. Y. Bordin Yamashita, & J. P. R. R. Leite. (2023). Coffee disease classification at the edge using deep learning. *Smart Agricultural Technology*, 4, 100183. DOI: 10.1016/j.atech.2023.100183
- [32] U. Sanath Rao et al. (2021). Deep learning precision farming: Grapes and mango leaf disease detection by transfer learning. *Global Transitions Proceedings*, 2(2), 535–544. DOI: 10.1016/j.gltp.2021.08.002
- [33] S. Anwarul, M. Mohan, & R. Agarwal. (2023). An unprecedented approach for deep learning assisted web application to diagnose plant disease. *Procedia Computer Science*, 218, 1444–1453. DOI: 10.1016/j.procs.2023.01.123
- [34] G. Geetharamani, & A. P. J. (2019). Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Computers and Electrical Engineering*, 76, 323–338. DOI: 10.1016/j.compeleceng.2019.04.011
- [35] U. P. Singh, S. S. Chouhan, S. Jain, & S. Jain. (2019). Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease. *IEEE Access*, 7, 43721–43729. DOI: 10.1109/ACCESS.2019.2907383
- [36] J. Chen, Q. Liu, & L. Gao. (2019). Visual tea leaf disease recognition using a convolutional neural network model. *Symmetry*, 11(3). DOI: 10.3390/sym11030343
- [37] B. Liu, Z. Ding, L. Tian, D. He, S. Li, & H. Wang. (2020). Grape leaf disease identification using improved deep convolutional neural networks. *Frontiers in Plant Science*, 11, 1–14. DOI: 10.3389/fpls.2020.01082
- [38] J. A. Pandian et al. (2022). A five convolutional layer deep convolutional neural network for plant leaf disease detection. *Electronics*, 11(8). DOI: 10.3390/electronics11081266
- [39] P. S. Thakur, P. Khanna, T. Sheorey, & A. Ojha. (2022). Explainable vision transformer enabled convolutional neural network for plant disease identification: PlantXViT. *arXiv preprint*. DOI: 10.48550/arXiv.2207.07919
- [40] S. T. Y. Ramadan, T. Sakib, R. Jahangir, & S. Rahman. (2024). Maize leaf disease detection using vision transformers and CNN-based classifiers: Comparative analysis. *Smart Innovation, Systems and Technologies*, 376, 513–524. DOI: 10.1007/978-981-99-7711-6_41
- [41] W. Abdullah, C. Kim, & Y. Nam. (2024). A fresh look at tomato leaf disease recognition using vision transformers. *Optimization in Agriculture*, 1, 105–114. DOI: 10.61356/j.oia.2024.1274
- [42] M. H. Alnamoly, A. A. Hady, S. M. Abd El-Kader, & I. El-Henawy. (2024). FL-ToLeD: An improved lightweight attention

- convolutional neural network model for tomato leaf diseases classification for low-end devices. *IEEE Access*, 12, 73561–73580. DOI: 10.1109/ACCESS.2024.3401733
- [43] M. Zhang et al. (2025). Dual-attention-enhanced MobileViT network: A lightweight model for rice disease identification in field-captured images. *Agriculture*, 15(6), 1–22. DOI: 10.3390/agriculture15060571
- [44] M. R. Tonmoy, M. M. Hossain, N. Dey, & M. F. Mridha. (2025). MobilePlantViT: A mobile-friendly hybrid ViT for generalized plant disease image classification. *arXiv preprint*, 1–8. Available: <http://arxiv.org/abs/2503.16628>
- [45] S. K. Upadhyay, & A. Kumar. (2022). A novel approach for rice plant diseases classification with deep convolutional neural network. *International Journal of Information Technology*, 14(1), 185–199. DOI: 10.1007/s41870-021-00817-5
- [46] X. Li, & S. Li. (2022). Transformer help CNN see better: A lightweight hybrid apple disease identification model based on transformers. *Agriculture*, 12(6). DOI: 10.3390/agriculture12060884
- [47] S. Nishankar, V. Pavindran, T. Mithuran, S. Nimishan, S. Thuseethan, & Y. Sebastian. (2025). ViT-RoT: Vision transformer-based robust framework for tomato leaf disease recognition. *AgriEngineering*, 7(6), 185. DOI: 10.3390/agriengineering7060185
- [48] M. A. Hasan et al. (2025). Mulberry leaf disease detection by CNN-ViT with XAI integration. *PLOS ONE*, 20(6), 1–29. DOI: 10.1371/journal.pone.0325188
- [49] G. Shandilya, S. Gupta, H. G. Mohamed, S. Bharany, A. U. Rehman, & S. Hussien. (2025). Enhanced maize leaf disease detection and classification using an integrated CNN-ViT model. *Food Science & Nutrition*, 13(7). DOI: 10.1002/fsn3.70513
- [50] A. Hassani, S. Walton, J. Li, S. Li, & H. Shi. (2023). Neighborhood attention transformer. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6185–6194. DOI: 10.1109/CVPR52729.2023.00599
- [51] A. Hassani. (2023). Neighborhood attention: Dynamic restriction of self-attention. University of Oregon Area Exam Report. Available: <https://www.cs.uoregon.edu/Reports/AREA-202307-Hassani.pdf>
- [52] A. Hassani, W. Hwu, & H. Shi. (2024). Faster neighborhood attention: Reducing the $O(n^2)$ cost of self-attention at the threadblock level. *arXiv preprint*. DOI: 10.48550/arXiv.2403.04690
- [53] A. Ouamane et al. (2025). Optimized vision transformers for superior plant disease detection. *IEEE Access*, 13, 48552–48570. DOI: 10.1109/ACCESS.2025.3547416
- [54] S. Duhan, P. Gulia, N. S. Gill, & E. Narwal. (2025). RTR_Lite_MobileNetV2: A lightweight and efficient model for plant disease detection and classification. *Current Plant Biology*, 42, 100459. DOI: 10.1016/j.cpb.2025.100459
- [55] O. Jouini, M. O. E. Aoueilayine, K. Sethom, & A. Yazidi. (2024). Wheat leaf disease detection: A lightweight approach with shallow CNN based feature refinement. *AgriEngineering*, 6(3), 2001–2022. DOI: 10.3390/agriengineering6030117
- [56] A. Dosset et al. (2025). Cassava disease detection using a lightweight modified soft attention network. *Pest Management Science*, 81(2), 607–617. DOI: 10.1002/ps.8456
- [57] A. Chakrabarty, S. T. Ahmed, M. F. U. Islam, S. M. Aziz, & S. S. Maidin. (2024). An interpretable fusion model integrating lightweight CNN and transformer architectures for rice leaf disease identification. *Ecological Informatics*, 82, 102718. DOI: 10.1016/j.ecoinf.2024.102718
- [58] E. Zhang, N. Zhang, F. Li, & C. Lv. (2024). A lightweight dual-attention network for tomato leaf disease identification. *Frontiers in Plant Science*, 15, 1–18. DOI: 10.3389/fpls.2024.1420584

Authors' Profiles



Rithambara Rajput received her Bachelor of Engineering degree in Electronics and Communication Engineering from Visvesvaraya Technological University (VTU) in 2019 and her M.Tech. degree in VLSI Design in 2021. She is currently pursuing a Ph.D. at KLE Technological University's Dr. M. S. Sheshgiri College of Engineering and Technology, Belagavi, India. Her research interests include image processing, deep learning, computer vision, lightweight neural network design, plant disease detection, and the application of advanced computing techniques to agricultural and engineering problems.



Dr. Suneeta V. Budihal earned her Ph.D. in Electronics and Communication Engineering from KLE Technological University, Hubballi, in 2018. Her major field of study includes wireless communication systems, machine learning applications, and engineering education methodologies, with a focus on practical and interdisciplinary problem solving. She is currently a Professor and Head of the School of Electronics and Communication Engineering at KLE Technological University, Hubballi, a position she has held since 2023. With over two decades of academic experience, her work spans LTE and cooperative communication systems, CNN-based solutions for agriculture and vision applications, and the adoption of experiential and reflective learning practices in engineering education.



Dr. Saroja V. Siddamal received her Ph.D. in VLSI Signal Processing from Jawaharlal Nehru Technological University (JNTU), Hyderabad, in 2013. Her academic specialization lies in VLSI design, digital signal processing, and high-performance arithmetic architectures for signal processing applications. She is presently a Professor and Head of the School of Electrical and Electronics Engineering at KLE Technological University, Hubballi. She has been associated with the institution since 2002 and has contributed extensively through research publications, academic leadership, activity-based teaching practices, and entrepreneurial initiatives in the energy systems domain



Dr. Dattaprasad A. Torse earned his Ph.D. from Visvesvaraya Technological University (VTU) in Biomedical Signal Processing, with a major focus on EEG signal analysis for healthcare applications. His academic background also includes postgraduate studies in Digital Electronics, strengthening his expertise in intelligent signal analysis and machine learning. He is currently serving as Professor and Head of the Department of Electronics and Communication Engineering at KLE Technological University, Dr. M. S. Sheshgiri Campus, Belagavi, since January 2022. His work centers on biomedical signal processing, machine learning and deep learning models for diagnostic systems, and intelligent wireless and semantic communication networks.

How to cite this paper

Rithambara Rajput, Suneeta Budihal, Saroja Siddamal, Dattaprasad Torse, "A Lightweight Convolutional Neural Network with Neighbourhood Attention and a 100-Category Dataset for Plant Disease Detection", International Journal of Engineering and Manufacturing (IJEM), Vol.16, No.3, pp.193-217, 2026. DOI:10.5815/ijem.2026.03.12