Sustainable Food Technology

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Sustainable Food Technol.*, 2026, 4, 985

L-Net: a lightweight CNN framework for sustainable multicrop leaf disease detection and classification on edge devices

R. Deepa, [a] Midhun P. Mathew, *[b] S. Baskar [c] and Abubeker K. M *[d]

The early diagnosis of plant leaf diseases is crucial in the sustainable management of agriculture as it minimises crop damage and reduces the use of pesticides. This paper presents Leaf Net (L-Net), a new lightweight convolutional neural network for the detection and classification of leaf diseases in apple, bell pepper, and grape. The model includes depthwise separable convolutions within the layers of the model to capture features more efficiently, an ensemble activation function to improve non-linearity of the output, and a Modified Adamax optimiser to improve convergence. The datasets used include publicly available repositories as well as custom annotated images, which were later pre-processed and augmented to enhance generalizability. A plant-wise split cross-validation approach was used in training and evaluation, along with the partitioning scheme to avoid data leakage and increase the practical applicability of the results. L-Net obtained a classification accuracy of 99.8% and AUC score of 1.00. Though the variability in precision-recall metrics suggests that improvements are needed in performance at the class level, L-Net was shown to be compatible with low-power devices such as Raspberry Pi and NVIDIA Jetson Nano edge platforms, which proved its feasibility for detection in the field. Moreover, this model facilitates the diagnosis of plant diseases in a timely and precise manner and helps in the accurate application of pesticides and the management of crops. This, in turn, fosters the adoption of sustainable agricultural practices. Additional research focuses on cross-crop studies and real-world scaling of L-Net to enhance its model robustness.

Received 8th May 2025
Accepted 18th November 2025

DOI: 10.1039/d5fb00191a

rsc.li/susfoodtech

Sustainability spotlight

This research contributes to sustainable agriculture by presenting L-Net, a lightweight and highly accurate deep learning model for the early identification and classification of leaf diseases in bell pepper, grape, and apple plants. By enabling real-time, low-computation disease detection on resource-constrained devices, L-Net empowers farmers with cost-effective, scalable, and autonomous solutions for crop monitoring. The architecture's integration of depthwise separable convolutions minimizes energy consumption and enhances processing efficiency, making it suitable for deployment in remote and rural farming environments. This innovation supports precision agriculture by reducing chemical usage through timely disease intervention and improving yield with minimal environmental impact—thereby aligning with the United Nations Sustainable Development Goals (SDGs) related to Zero Hunger (SDG 2) and Responsible Consumption and Production (SDG 12).

1. Introduction

For the crops on which these initiatives document the world's food supply and achieve food security, there are consequences and impacts of various diseases that can affect yield and value. Apple, grape, and bell pepper crops are among the most common instances of diseased vegetation where the symptoms and diseases affect the leaves and generate losses in value. Timely detection of these symptoms is critical for planning the rational implementation of management initiatives to control losses. The classical and most common diagnostic approaches rely on human observations, which are slow, tedious, and error-prone. For many of the most common symptoms and leaf diseases, there is now the possibility of automated and accurate diagnoses. The primary algorithms of interest are based on deep learning (DL) and convolutional neural network (CNN) approaches that have proven to excel in various image classification problems. The most potent and resource-consuming algorithms that are being reported in the literature are

[a]*Assistant Professor, Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur Campus, Chennai, Tamil Nadu, India. E-mail: deepar2@srmist.edu.in*

[b]*Assistant Professor, Department of Computer Science and Engineering, Amal Jyothi College of Engineering (Autonomous), Kanjirappally, Kerala, India. E-mail: midhunpmathew@amaljyothi.ac.in*

[c]*Assistant Professor, Department of Electronics and Communication Engineering, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India. E-mail: connectbaskar@gmail.com*

[d]*Associate Professor, Department of Electronics and Communication Engineering, Amal Jyothi College of Engineering (Autonomous), Kanjirappally, Kerala, India. E-mail: kmabubeker82@gmail.com*

unrealistic for small farms to perform in real time and to rely on edge computing devices. Most of the research ignores the real-world problems of variable illumination, occlusions, and terrain distortions for leaf disease classification in closed and controlled environments.

This research presents a neural network architecture, L-Net, implemented from scratch, which is lightweight and designed to detect and differentiate foliar diseases, frontal and dorsal, for apple, grape, and bell pepper crops from a close range or ground level distance. The L-Net model is designed for rapid deployment to real agricultural fields due to its lightweight, power-efficient, and highly accurate capabilities, thus outshining competing models. The model underwent extensive testing, spanning real-life conditions, and is reported to give over 98% accurate results on training and testing validity, which stands to prove its effectiveness. This contribution can be summarised in the following key points:

- Developing and fine-tuning a new lightweight CNN architecture (L-Net)
- Trained from the ground level to achieve the highest accuracy in multiple crops, apple, grape, and bell pepper, for leaf disease detection.

Detection of frontal and dorsal foliar diseases was conducted under real-world conditions concerning occlusions and variations in illumination. The proposed architecture is highly proficient and surpasses the efficiency of deep learning models deployed for wider applications.

1.1 Novelty of L-Net

Models such as MobileNet and EfficientNet are applicable in resource-limited settings, but these are built for general image classification and depend on transfer learning. These architectures contain redundant parameters for tasks like classifying plant diseases, where the recognisable patterns of disease are sparse and highly detailed. Conversely, L-Net incorporates depthwise separable convolutions to reduce computation for multicrop leaf disease classification, as well as an ensemble activation function, Gaussian error linear unit (GELU), and leaky rectified linear unit (leaky ReLU) to improve feature learning for the numerous and diverse disease patterns. Furthermore, L-Net employs a Modified Adamax to enhance learning stability over severely unbalanced agricultural datasets. Unlike MobileNet and EfficientNet, this architecture focuses on edge device use, tuned for low-latency, real-time execution in rural or field contexts, making it more innovative and more sustainable for farming.

1.2 Justification for crop selection

Economic importance, the availability of data, and disease diversity make bell pepper, grape, and apple relevant for the research. These crops are essential horticultural crops grown extensively in temperate and subtropical regions. Apple and grape are among the top fruit crops, in production and export value, whereas bell pepper is an important vegetable crop with a high demand and perishability, along with a propensity for particular diseases. Importantly, each of these crops is exposed to numerous visually identifiable foliar diseases such as apple scab, grape black measles, and bell pepper bacterial spot, which aids in classification using image processing. Moreover, these crops are well represented in publicly available datasets and are often targeted by agricultural disease surveillance programs. This combination of plant economic importance, diversity of diseases, availability of datasets, and accessibility makes them suitable for developing and validating a generalised, lightweight CNN model for diagnosing plant leaf diseases.

1.3 Sustainability relevance of the research

This work aids in achieving the goals of sustainable food systems by facilitating early, precise, and scalable detection of crop diseases, thereby preventing unnecessary application of chemical pesticides and minimising losses in yield. Automated leaf disease identification enables the selective use of fungicides and bactericides. Moreover, the designed L-Net architecture is light enough to be implemented on low-cost edge devices, such as Raspberry Pi and Jetson Nano. This is important to low-resource farming communities, which are limited to minimal cloud computation and agronomic knowledge. Equipping smallholder farmers with real-time devices enables them to make quick, informed, and confident decisions, which substantially decrease waste and improve efficiency, thereby promoting resource-efficient and climate-resilient agriculture.

The rest of the paper is organised as follows: Section 2 discusses the literature on plant disease classification using deep learning techniques. Section 3 describes the model development, followed by the methodology in Chapter 4, which entails L-Nets's architecture and dataset construction for the model. Section 5 describes the experimental setup and analysis of the results, while Section 6 presents model analysis with an accompanying discussion. Finally, Section 7 offers conclusions and outlines directions for future research.

2. Literature review

There has been considerable effort towards employing deep learning and machine learning (ML) models to detect plant diseases. Sivaganesh *et al.*[1] used statistical and machine learning techniques to distinguish varying degrees of Fusarium wilt on tomatoes by remote hyperspectral sensing. Padhi *et al.*[2] used simulated thermal imaging in a hybrid CNN model to improve the diagnosis of paddy leaf diseases. Dhoundiyal *et al.*[3] developed a method for plant disease diagnosis using a progressive hierarchical model to achieve enhanced accuracy and efficiency. Classifying various diseases has heavily relied on novel DL architectures. Bhookya *et al.*[4] created and trained a DBESeriesNet model to classify crop leaf diseases. Hanif *et al.*[5] performed a comparative analysis of image classification using ResNet, Inception-v3, and support vector machine (SVM) for plant diseases. In addition to crop leaf diseases, the authors designed a semi-automated system for grape leaf disease, which improved accuracy and efficiency in agricultural practices.[6] Moreover, CNNs have been commonly used to detect plant diseases. Nain *et al.*[7] analysed the application of some colour

space models in CNN-based recognition of plant diseases. Bahrami *et al.*[8,9] used transfer learning (TL) techniques; they conducted a comparative study on recognising diseases in tomato leaves. Kadam *et al.*[10] also suggested an approach of automatic image detection targeted at diagnosing disease in apple fruiting plants. In ref. 11, Khan *et al.* developed a new dense-inception architecture with attention mechanisms to improve the classification of plant diseases.

Other developments and research have targeted real-time models and graphics processing unit (GPU) accelerated ones. Rahman *et al.*[12] developed a system for monitoring and detecting plant diseases that works in real-time and relies on deep learning techniques. Among other things, Wang *et al.*[13] were concerned with the isolation and identification of the causative agent of gummy stem blight disease of cucumber, contributing to the understanding of plant pathology. Alongside these developments, transformer models and federated learning methods were also crafted. Chai *et al.*[14] proposed the PlantAIM model, which combines global attention and local features for identifying plant diseases. Hari and Singh[15] presented an adaptive method of knowledge transfer using federated deep learning for plant disease detection with privacy and efficiency concerns.

Liu *et al.*[16] developed an advanced YOLOv5-based model, achieving 92.7% mAP in apple leaf disease detection. In a recent research study, a novel mobile-optimised lightweight model YOLOv8n–GGi is designed for an apple leaf disease detection model designed to work in natural environments.[17] The model was optimised through the application of GhostConv, C3Ghost, GAM, and BiFPN modules, where it achieved 86.9% mAP. Advanced DL models, particularly with Swinv2-Base, are achieving 100% accuracy in early diagnosis and identification, thus highlighting the ease of integration of deep learning in agriculture.[18] In ref. 19, real-time grape leaf disease detection using the MobileNetV3Large model is developed and deployed on the NVIDIA Jetson Nano edge platform. The model was able to achieve 99.66% training accuracy and 99.42% test accuracy. In ref. 20, spanning various ML and DL frameworks designed for early plant leaf disease detection, the paper emphasises the EfficientNet family of models, which attained 98.12% accuracy for image classification at a modest computational cost. Sustainable agriculture and food security hinge on accurate leaf disease detection. An ANFIS-integrated CNN model with local binary pattern (LBP) features was used for enhanced detection of bell pepper leaf diseases. With the LBP, the model achieved exceptional accuracy, surpassing 99%, revealing its potential for dependable agricultural applications.[21] Table 1 shows a summary of the literature.

In contrast, the developed L-Net model for embedded systems considers hardware constraints and spatial contextual domain patterns. L-Net justifies this by completing the primary accuracy target for the embedded domain while using depthwise separable convolution, a novel ensemble activation function, and a customised optimiser. Apart from this, most literature ignores performance under the multi-crop, multi-disease, and class imbalance conditions. This work aims to close that gap using strategic augmentation, balanced partitioning, and a class-level evaluation framework. Evidence for the practical nature of this work is provided in Section 5, which includes comparisons of inference times for L-Net and baseline CNN, MobileNet, and EfficientNet architectures.

3. Development of L-Net architecture

Training L-Net the first time without any pre-trained weights or transfer learning was to ensure that the model fully adapted to the specific characteristics of the images of diseased plant leaves, because there are considerable differences between these images and general datasets like ImageNet. Also, training from scratch has allowed the model to learn representations of specific features such as the textures, colours, and diseased patterns specific to crops like bell peppers, grapes, and apples. Finally, it provided more design freedom in customisation, such as adding ensemble activation functions and an alternative optimisation approach, which otherwise would be hard to implement when transfer learning from rigid baseline models.

Table 1 Comparative summary of existing models for plant disease classification

Sl. No.	Model/Study	Dataset used	Advantage	Disadvantage	Accuracy (%)	Precision (%)	Research gap
1	MobileNetV2 (ref. 5)	Plant village	Efficient on mobile and edge devices	Relies on transfer learning; lower class-specific precision	98.2	96.3	Limited crop-specific optimisation
2	EfficientNetB0 (ref. 8)	Plant village	Balanced accuracy and model size	Higher inference latency on low-end devices	98.6	97.2	Not optimised for real-time agricultural field use
3	DBESeriesNet (ref. 4)	Plant village	Tailored for leaf disease classification	Moderate model size; lacks edge evaluation	98.0	96.9	No hardware constraint analysis or field testing
4	DSC-TransNet (ref. 11)	Plant village	GPU-enabled, real-time detection	Needs expensive hardware, not practical for low-resource areas	97.4	96.5	High deployment cost and energy demand
5	Dense-inception + attention (ref. 10)	Plant village	Improved spatial attention and feature localisation	High complexity and latency	97.8	97.0	No sustainability or deployment framework was discussed

3.1. Block for extraction of features

The initial part of L-NET consists of depthwise separable convolution layers, enabling a considerable decrease in computational cost without losing important spatial characteristics. A depthwise convolution filter is applied by utilising only one convolutional filter for each input channel, followed by pointwise feature extraction:

$$X' = (W_{\mathrm{d}} \times X) \times W_{\mathrm{p}} \quad (1)$$

In this case, W_{d} is the depthwise filter, W_{p} is the pointwise filter, and X is the input feature map. Batch normalisation is performed to stabilise the learning process, followed by an ensemble activation function, which is a combination of GELU and Leaky ReLU:

$$f(x) = \tfrac{1}{2}(\mathrm{GELU}(x) + \mathrm{LeakyReLU}(x,\alpha)) \quad (2)$$

where GELU is defined as:

$$\mathrm{GELU}(x) = x\Phi(x), \text{where}\Phi(x) = 1/2\left(1 + \tan h\left(\sqrt{(2/\pi)}\,(x + 0.044715x^3)\right)\right) \quad (3)$$

And Leaky ReLU is defined as:

$$\mathrm{LeakyReLU}(x,\alpha) = \max(\alpha x, x) \quad (4)$$

The architecture consists of three main convolutional blocks, each containing two depthwise separable convolution layers with 32, 64, and 128 filter sizes, respectively. Batch normalisation with renormalisation follows each block for progressive weight updating, and max pooling with a stride of 2 reduces the spatial dimensions:

$$X_{l+1} = \max(X_l, 2 \times 2) \quad (5)$$

where X_l represents the feature maps at layer l, and max pooling retains the most prominent feature responses while discarding irrelevant information. A 1×1 convolutional layer with 32 filters is introduced before classification to enhance channel-wise feature representation without affecting spatial dimensions:

$$X' = W \times X + b \quad (6)$$

In eqn (6), W is the 1×1 filter, X is the input, and b is the bias term. This transformation helps refine extracted patterns before feeding them into the classification head. The final stage includes global average pooling:

$$X_{\mathrm{avg}} = (1/H \times W)\Sigma\Sigma X_{ij} \quad (7)$$

In eqn (7), H and W are the height and width of the feature map. The pooled feature vector is flattened and passed through two fully connected layers (512 and 256 neurons), each using the ensemble activation function and L2 regularisation to prevent overfitting:

$$L_{\mathrm{reg}} = \lambda||W||^2 \quad (8)$$

A dropout of 0.5 and 0.3 is applied to improve generalisation. The final softmax output layer produces probability distributions for two classes, using the categorical cross-entropy loss:

$$L = -\Sigma y_i \log(\hat{y}_i) \quad (9)$$

In eqn (9), y_i is the true label, $\hat{y}_i$ is the predicted probability, and C is the number of classes. L-Net is optimised using a Modified Adamax optimiser, which refines the standard Adamax update rule by introducing an additional moving average term. This optimisation technique includes the following equations:

$$m_t = \beta_1 m_{(t-1)} + (1 - \beta_1) g_t \quad (10)$$

$$v_t = \max(\beta_2 v_{(t-1)}, |g_t|) \quad (11)$$

$$v_t = v_{(t-1)} - \mu g_t + \varepsilon v_t \quad (12)$$

$$\theta_t = \theta_{(t-1)} - \eta(m_t/(\hat{v}_t + \varepsilon)) \quad (13)$$

In eqn (10)–(13) g_t is the gradient, β_1 and β_2 are momentum coefficients, and η is the learning rate. A detailed comparative analysis between the proposed L-Net and state-of-the-art lightweight CNN architectures (MobileNet and EfficientNet) is provided in Table 2.

Table 2 Comparison of the developed L-Net with lightweight CNN architectures, such as MobileNet and EfficientNet

Feature/Model	MobileNetV2	EfficientNetB0	Proposed L-Net
Primary purpose	Generic image classification	Generic image classification	Crop-specific plant leaf disease classification
Training strategy	Transfer learning	Transfer learning	Trained from scratch
Activation function	ReLU6	Swish	Ensemble (GELU + leaky ReLU)
Optimizer	Adam	RMSProp	Modified Adamax
Model parameters	~3.4 million	~5.3 million	~1.9 million
FLOPs (multiply-add ops)	~300 million	~390 million	~36 million
Hardware target	Edge/Cloud	Edge/Cloud	Edge-focused (Jetson Nano, and Raspberry Pi)
Domain adaptability	Moderate	Moderate	Highly tailored for agricultural datasets
Suitability for real-time use	Medium	Medium	Low latency and fast inference
Sustainability relevance	Not explicitly addressed	Not explicitly addressed	Explicitly aligned with smart farming and sustainability

Table 3 Layer-wise architecture details of the developed L-Net model

Layer type	Output shape	Kernel/Stride	Filters	Params	FLOPs (M)
Input	256 × 256 × 3	—	—	0	0
Depthwise conv 1	256 × 256 × 32	3 × 3/1	32	320	15.1
Pointwise conv 1	256 × 256 × 32	1 × 1/1	32	1024	4.2
Max pooling	128 × 128 × 32	2 × 2/2	—	0	0.8
Depthwise conv 2	128 × 128 × 64	3 × 3/1	64	640	7.5
Pointwise conv 2	128 × 128 × 64	1 × 1/1	64	2048	8.1
Max pooling	64 × 64 × 64	2 × 2/2	—	0	0.4
Depthwise conv 3	64 × 64 × 128	3 × 3/1	128	1280	6.2
Pointwise conv 3	64 × 64 × 128	1 × 1/1	128	8192	12.4
Max pooling	32 × 32 × 128	2 × 2/2	—	0	0.2
1 × 1 bottleneck conv	32 × 32 × 32	1 × 1/1	32	4096	2.1
Global avg pooling	1 × 1 × 32	—	—	0	0.03
FC layer 1	512	—	—	16 896	0.06
Dropout (0.5)	512	—	—	0	0
FC layer 2	256	—	—	131 072	0.05
Dropout (0.3)	256	—	—	0	0
Softmax output	6 (classes)	—	—	2560	0.02
Total	—	—	—	167 128	56.9

To illustrate the lightweight nature of L-Net, Table 3 presents a full layer-by-layer summary showing output size, parameter count, and estimated FLOPs for each stage in the pipeline. Overall, the model uses roughly 167 000 parameters and executes around 56.9 million FLOPs per forward pass. This slim architecture lowers memory footprint and accelerates inference speed, supporting the claim that L-Net is practical for real-time disease screening on edge hardware such as the Raspberry Pi 4B and Jetson Nano.

Fig. 1 describes a streamlined deep learning architecture designed for image classification, emphasising low computational requirements and suitability for real-time or edge deployment scenarios. The system processes an input image of size 256 × 256 × 3. It begins with feature extraction through a series of depthwise separable convolution layers, which drastically reduce the number of learnable parameters while preserving representational capacity. The initial convolution block contains two depthwise separable convolutions of 32 filters each, which is followed by batch normalisation to stabilise and improve training dynamics. This scheme is duplicated with increasing filter sizes of 64 and 128, followed by batch normalisation and 2 × 2 max pooling layers for the reduction of spatial dimensions. The transition from convolutional layers to fully connected layers is aided by a 1 × 1 conv2D layer with 32 filters, which performs some degree of dimensionality reduction, and is followed by a 7 × 7 average pooling layer, which integrates some spatial characteristics. The resulting feature map is subsequently flattened and passed to two fully connected layers with 512 and 256 neurons, respectively. An ensemble of activation functions is employed, comprising a number of nonlinear functions, with the purpose of enhancing generalisation. To further mitigate the effects of overfitting, dropout layers are utilised with probabilities of 0.5 and 0.3 on the dense layers. The final dense layer has a Softmax activation specifically designed for two-class outputs. The Modified Adamax Optimiser is used, which combines the standard modified optimisers with a custom-designed learning rate algorithm that adapts to a large range of datasets to optimise convergence and performance.

4. Methodology

The stages involved in identifying plant leaf diseases using L-Net are outlined in Fig. 2. These stages are as follows: capturing images, compiling a dataset, dataset segmentation, training an L-Net model, classifying diseases, and calculating overall performance metrics. The first of these stages involves obtaining leaf image data from various data archives, online repositories, or other sources. The next step is image augmentation, which comprises techniques such as rotation, flipping, scaling, changing colour, and several different methods to increase dataset variability. The complete image classification pipeline is depicted in Fig. 2. Initially, the process starts with the data source, which also encompasses the dataset, after which it is divided into three subsets: 70% is allocated for training, 20% for validation, and the remaining 10% for testing. The training and validation sets undergo image augmentation to enhance data diversity and refine the model's performance. The L-Net model, a lightweight convolutional neural network trained for the classification of plant diseases, is trained on the augmented dataset mentioned earlier. The model then undergoes hyperparameter optimisation, which adjusts critical learning parameters, including the learning rate, batch size, and dropout rate, to enhance performance.

Upon identifying the best configuration, the L-Net model is trained using the training dataset and subsequently assessed using the validation set. Evaluation of the model is then performed on an unbiased testing dataset. The model is evaluated on performance metrics that include accuracy, precision, recall, F1 score, and AUC during the final evaluation to demonstrate the efficacy of the proposed model. The implementation of this workflow in testing ensures achieving all of the following for the

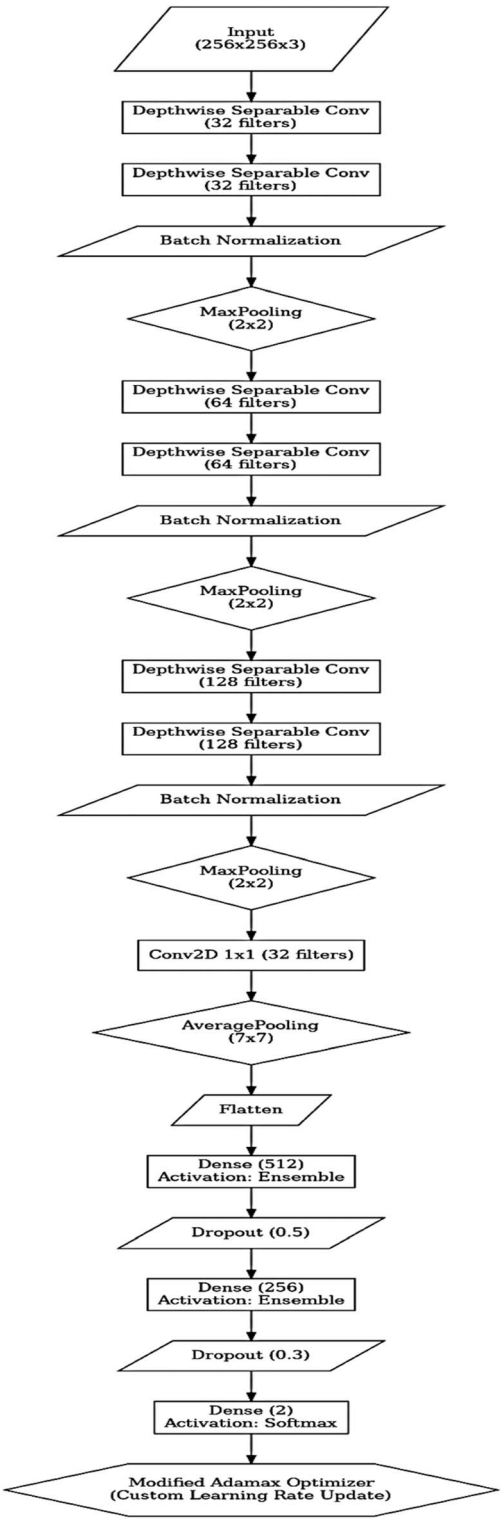

Fig. 1 Flow chart of the developed L-Net model.

developed models: reliability, robustness, reproducibility, and preparation for deployment/additional testing.

4.1 Dataset preparation (label validation and ground truth)

For the most reliable and scientifically valid form of disease labels, the image annotations came from one of two origin streams. These streams consist of expert-validated public collections from Kaggle[22,23] and GitHub repositories[24,25] and custom images evaluated by agronomists and plant pathologists. For newly obtained images, an initial set of labels was developed by the researchers based on visual symptoms and standardised protocols for plant disease. These were then reviewed by two additional specialists from the department of agricultural science at a nearby regional university. The reviewers were required to reach consensus on the disease class for each of the sample images. Any ambiguous images, such as those displaying overlapping symptoms or evident environmental damage, were excluded. This was done in order to not compromise the quality of the remaining images. These multi-layered annotations provide the data set with a reliable ground truth that reflects real-world signs of the disease and allows for the model to be trained and tested effectively at subsequent stages of this research.

4.2 Dataset preparation

To minimize data leakage and to objectively evaluate the model, we opted for a plant-wise splitting method rather than the traditional image-wise split. As for this method, we grouped all the images of a specimen, regardless of the different angles, lighting conditions, or magnifications, and assigned the entire collection to one of the three sets: training, validation, or testing. This configuration of similar images into one phase allows us to mitigate the risk of artificially inflating performance metrics, because the model never sees augmented duplicates of a specimen at different stages of training. This plant-wise configuration was uniformly applied to both the original and augmented datasets, allowing the final test to accurately assess the generalisation of the system to completely new specimens.

4.3 Image pre-processing

The focal point of this study is the dataset obtained using photographs of plant leaves, where each leaf is calibrated to a resolution of 256 by 256 pixels. While standardising the image improves the leaf dataset's uniformity and removes unnecessary computational differentiation, it unfortunately complicates the uploading pre-processing steps. Furthermore, standardising the image could have hindered the detection span of the model in terms of lesions that are smaller in size or modestly visible. In order to extract conclusive information, the model's performance was assessed by determining the accuracy of the model in original and resized images. The conclusion drawn from our results suggests that the resizing of the image does not hurt the model's performance concerning the plant leaf images. The sample image of this dataset is illustrated in Fig. 3.

Table 4 summarises the number of pictures for the various classifications of plant leaf diseases according to plant types such as bell pepper, apple, and grape.

This research employs a dataset containing a total of 7828 images, which are labelled and span across three plant species: bell pepper, grape, and apple, considering both healthy and diseased leaves. For bell pepper, the dataset contains two classes:

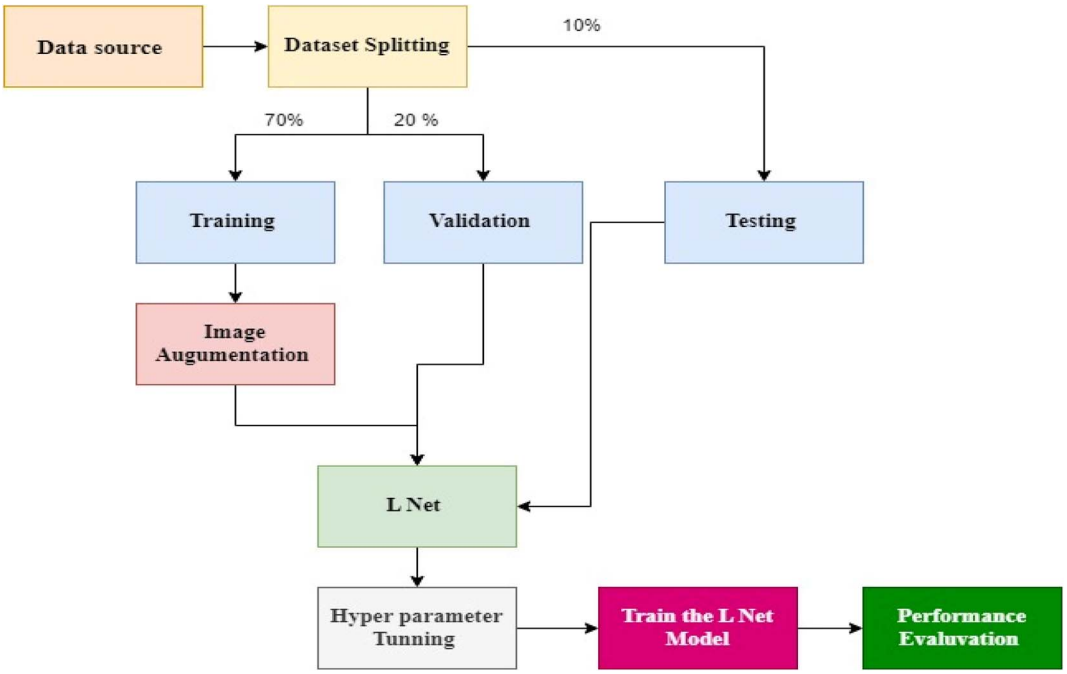

Fig. 2 The evaluation process of the L-Net model.

bacterial spot (797 images) and healthy (1183 images), totalling 1980 images. The grape category includes four classes: black rot (944 images), leaf spot (861 images), black measles (1107 images), and healthy (399 images), totalling 3311 images. The apple leaves also consist of four classes: scab (504 images), black rot (497 images), cedar apple rust (220 images), and healthy (1316 images), totalling 2537 images. Overall, the dataset contains a reasonably balanced showcase of many different types of diseased leaves and healthy leaves. This assists in training the deep learning models for evaluating the classification of diseased leaves and is effective in the training process.

4.4 Image augmentation

To enhance the model's performance, the augmentation of image data is pivotal in reinforcing and generalising the model. Constructing new variations within the dataset allows the model to adapt to the different changing real-world conditions. The primary augmentation techniques in this study are:

4.4.1 Rotation range. This allows images to be rotated within the limits of ±40°, which is needed for variability.

4.4.2 Width shift range (WSR) and height shift range (HSR). This allows random horizontal movements for up to 20% of the image's width or vertical movements for up to 20% of the image's height.

4.4.3 Shear range (SR). This allows for a maximum angle of 0.2 radians of shear transformation of the image, altering its shape.

4.4.4 Zoom range (ZR). This allows for random zooming of the images in or out by a satisfactory margin of 20%.

4.4.5 Horizontal flip (HF). This introduces additional spatial diversity by randomly flipping images with a 50% likelihood.

4.4.6 Fill mode. This concerns the newly created pixels from the transformation, thus ensuring smooth image transitions.

4.4.7 Normalisation. This normalises pixel values into a set range to mitigate variance in predictions based on imaging conditions.

4.4.8 Cropping and resizing. This maintains uniform dimensions of images, focusing on the area of interest and eliminating some non-relevant background noise.

The model utilises augmentation techniques, which improve model performance on unseen data, as the model learns the different representations of the same image. Example images modified with augmentation techniques can be seen in Fig. 4. For the L-Net model, the LevelNet model training set was expanded and diversified with augmentation techniques comprising rotations, shifts, shearing, zooming, and flipping.

The augmented dataset overview, as well as its characteristics, is contained in Table 5. These pre-processing approaches are aimed at ensuring that the model attends to important features and not irrelevant distortions. This is important, as it helps to correct the classifier's misclassification. Moreover, many techniques prevent overfitting by encouraging the model to learn patterns instead of memorising the augmented training data. Integrating these pre-processing and augmentation methods enhanced classification accuracy by 6% to 9%, and there was a reduction in the number of incorrect identifications made relative to the unprocessed images. Furthermore, the increase in dataset size helped improve the precision and recall values of the model, especially in the minority classes, which helped the model to identify rare plant diseases confidently. All pre-processing, as expected, was quantitatively evaluated, and it was shown that normalisation and cropping, as integrated

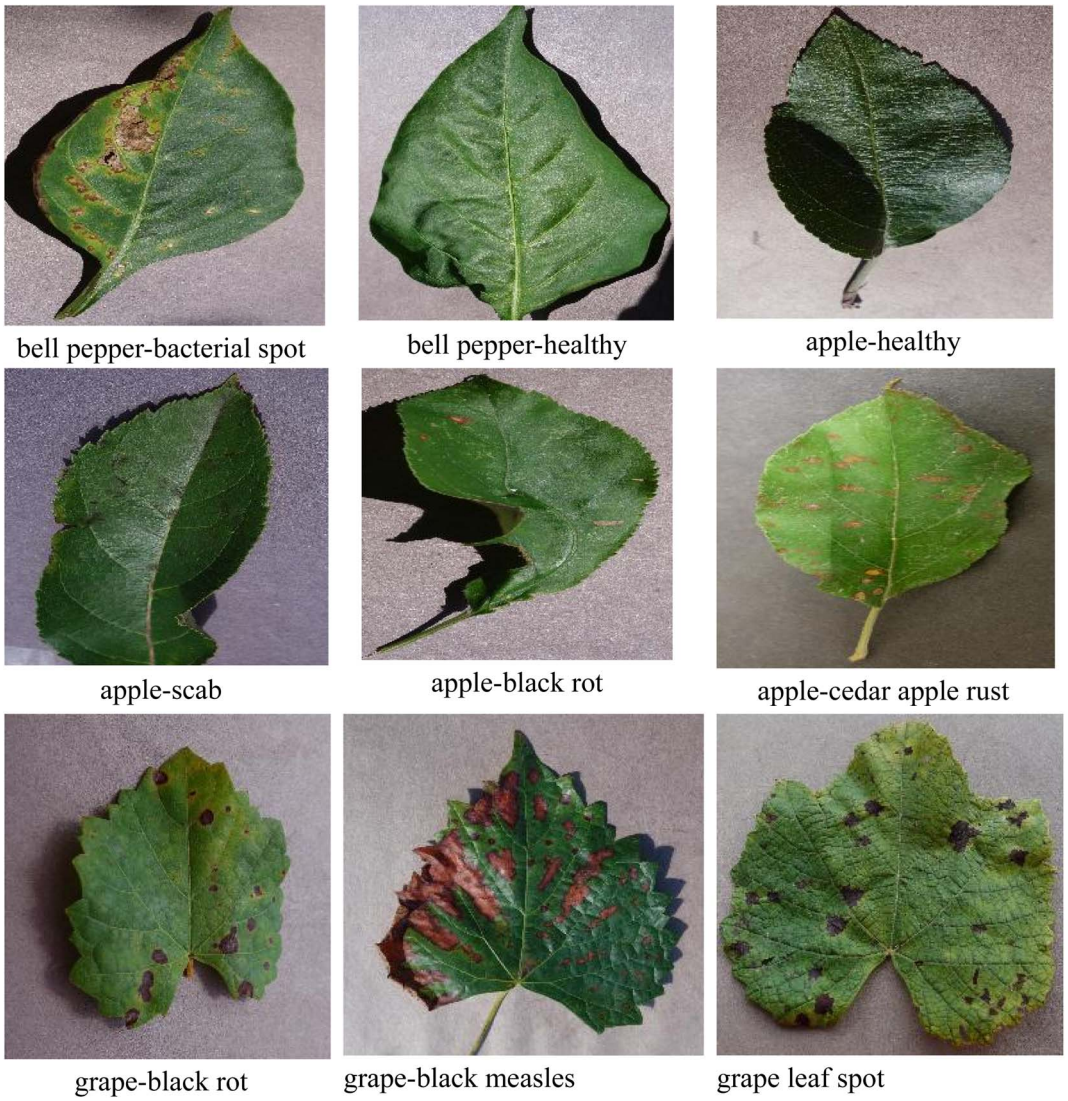

Fig. 3 Sample image of bell pepper, grape, and apple.

Table 4 Dataset summary for each class

Sl. No.	Type of plant	Disease type	No. of images
1	Bell pepper	Bacterial spot	797
		Healthy	1183
2	Grape	Black rot	944
		Leaf spot	861
		Black measles	1107
		Healthy	399
3	Apple	Scab	504
		Black rot	497
		Cedar apple rust	220
		Healthy	1316

workflows, were the most beneficial to the model in terms of feature clarity and performance.

For bell pepper, augmentation resulted in 5100 images for bacterial spot and 5200 for healthy samples. In the grape category, there are 6000 augmented images of black rot, 6300 of leaf spot, 6200 of black measles, and 6180 healthy images. In apple, the augmented dataset includes 5300 images of scab, 5200 of black rot, 5100 of cedar apple rust, and 5110 healthy images. This meticulous procedure supports balanced classification and improves the model's robustness by simulating different conditions of plant leaves in the real world.

4.5 Dataset preparation

The augmented dataset is systematically split into training (70%), testing (10%), and validation (20%) sets. This split facilitates model training and fair assessment of model performance:

4.5.1 Training set (70%). Used to derive insights from provided data and adjust model parameters.

4.5.2 Validation set (20%). This set offers the model a way to measure its performance on previously unseen data during the learning phase and assists in further calibrating hyperparameters such as the learning rate and regularisation rate.

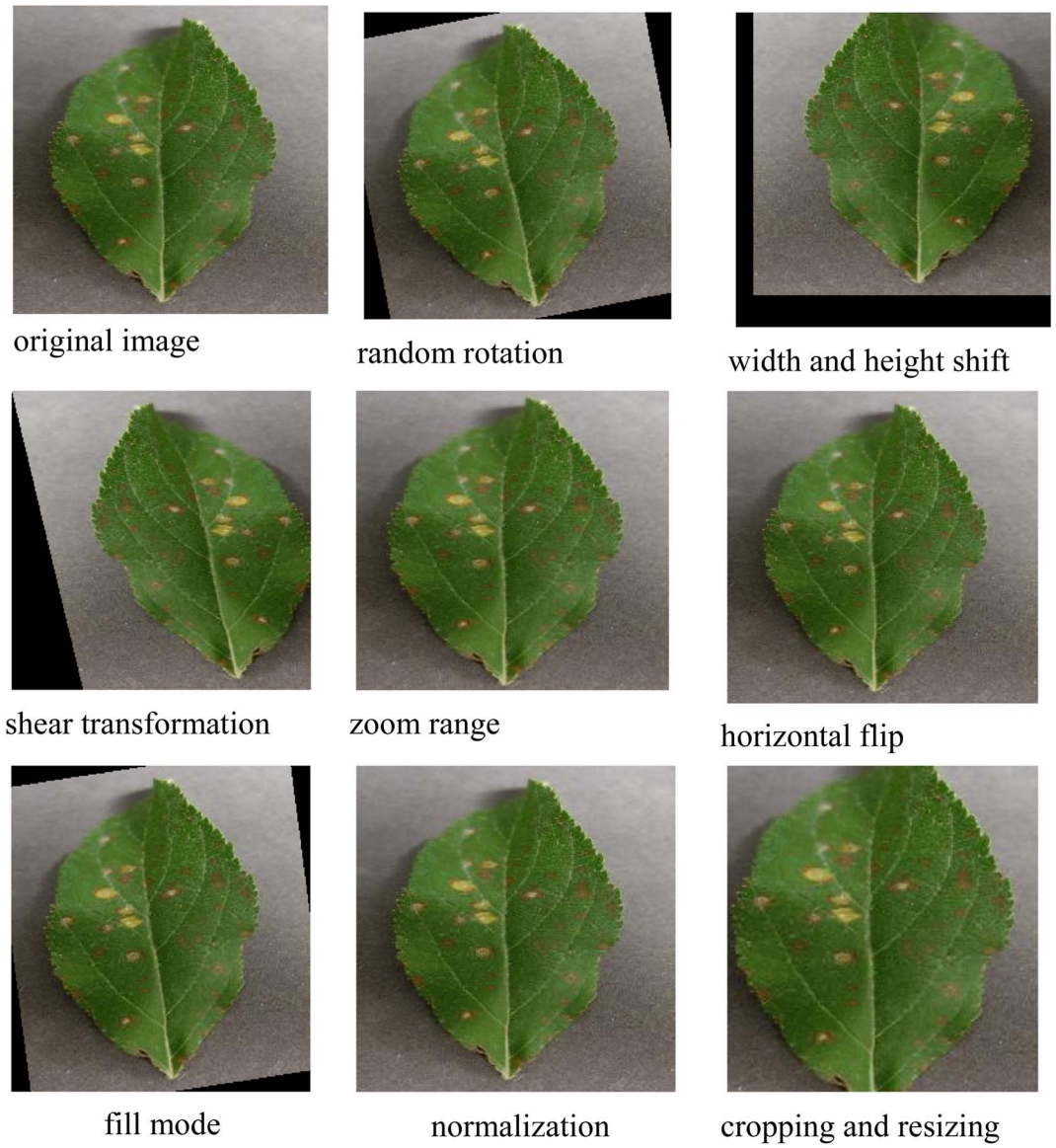

Fig. 4 Illustration of image transformations applied for augmentation.

4.5.3 Testing set (10%). This set assesses the final model's ability to use unknown data and measures its generalisation capability.

Table 5 Details of augmented images

Sl. No.	Type of plant	Disease type	No. of images
1	Bell pepper	Bacterial spot	5100
		Healthy	5200
2	Grape	Black rot	6000
		Leaf spot	6300
		Black measles	6200
		Healthy	6180
3	Apple	Scab	5300
		Black rot	5200
		Cedar apple rust	5100
		Healthy	5110
Total			55 690

Only the augmented images are included in the training set; for the original images, care was taken to avoid overlaps with the augmented images. This guarantees that no training, validation, or testing set shares any plants or leaves, and thus bias in evaluation results is avoided. To ensure data accuracy, a combination of automated and manual techniques is performed. The sample overlaps to ascertain a "clean", non-redundant, and unbiased superset and model for evaluation. To manage large data sets and quality control, we use TensorFlow data validation (TFDV), which is part of the TensorFlow Extended (TFX) suite. This step is vital for quality control of the set. The approach to the data set division is shown in Table 6. This validates unbiased objective criteria for model evaluation, thus preventing data pollution.

The dataset utilised for training, validating, and testing the L-Net model consisted of 55 810 images across three plant species: bell pepper, grape, and apple. For bell pepper, 10 300

Table 6 Dataset used in L-NET in training, validation, and testing

Sl. No.	Image type	Training	Testing	Validation	Total
1	Bell pepper	7210	1030	2060	10 300
2	Grape	17 360	2480	4960	24 800
3	Apple	14 497	2071	4142	20 710

images were collected, of which 7210 were used for training, 2060 for validation, and 1030 for testing. The dataset for grape contained 24 800 images, 17 360 of which were training images, 4960 for validation, and 2480 for testing. The apple dataset contained a total of 20 710 images, which included 14 497 training images, 4142 for validation, and 2071 for testing. This stratified split allows for balanced and sufficient learning to enhance the L-Net model's generalisation and classification proficiency across different plant species and disease types.

4.6 Reproducibility and averaging strategy

To improve replicability and reduce the impact of random variability during training, each condition was run five times with different random seeds. For every run, the evaluation metrics of accuracy, precision, recall, and F1 score were calculated and recorded on a held-out test split. The results presented include the average of the five trials along with standard deviations ($\pm\sigma$) for the metrics. This approach more accurately reflects the generalisation potential and stability of the model. For each run, hyperparameters, such as the learning rate and dropout rate, were set to the same values to ensure a fair evaluation.

5. Experimental results and performance evaluation

This experiment aimed to identify plant diseases related to bell pepper, grape, and apple leaves using L-Net. Image input was set to a standard size of 256 × 256 pixels. The training epochs ranged from 50 to 100 to achieve maximum precision during validation and testing. Various learning rates between 0.01 and 0.0001, plus different batch sizes, were implemented, but the optimal batch size to ensure efficacy was 9. The approach was evaluated comprehensively using multiple datasets for bell pepper, grape, and apple leaf diseases to ensure maximum comprehensiveness for the proposed model. The used hyperparameters with the L-Net model are given in Table 7.

The L-Net model was further trained with a specific set of hyperparameters tailored to achieve the best output and training efficiency to make any adjustments. The input image dimensions were set to 256, which provided both adequate resolution for feature extraction while remaining computationally efficient. Batch normalisation was active, which assisted in accelerating and stabilising training by normalising layer inputs. Convolutional layers included a stride value of 2 to downsample feature maps and reduce the spatial dimensions effectively. In addition, the architecture had a dense layer with 512 neurons that added a high-capacity representation before the output layer. Notably, the model used an ensemble activation function, which combines several nonlinear activation functions to improve the expressiveness and generalisation of the model. All these hyperparameters aided in striking a good balance in the trade-off between the model's complexity, stability during training, and accuracy during classification tasks.

Table 7 Details of the hyperparameters used in L-Net

Sl. No.	Hyperparameter	Setting
1	Input image size	256
2	Batch remoralization	True
3	Stride	2
4	Dense layer	512
5	Activation	Ensemble activation

The L-Net model was trained and evaluated on Google Colab Pro+, which provided a single NVIDIA A100 Tensor Core GPU with 40 GB of video memory. The virtual environment hosted 16 vCPUs and 85 GB of system RAM, and it ran Python 3.10, TensorFlow 2.12, and CUDA 11.8. Training proceeded for 100 epochs using a batch size of 9 and an initial learning rate of 0.001. During inference, the average prediction latency per image was about 36 milliseconds on the GPU, indicating that the model can classify inputs in near real time. This performance makes L-Net viable for cloud-assisted applications and lightweight edge devices, including Jetson Nano, Raspberry Pi paired with Coral TPU, or mobile platforms, where low latency and modest resource use are essential.

5.1 Ablation study on activation functions

An ablation study was carried out to test the new ensemble activation function, which blends GELU and Leaky ReLU, by replacing it with standard options: ReLU, Leaky ReLU, and GELU in isolation. To ensure a fair comparison, each trial used an identical network architecture and hyperparameter settings. Table 8 gives a complete summary and shows that the combined activation consistently surpasses all single functions on every central performance metric. In particular, L-Net reached a peak accuracy of 99.8% and precision of 99.7% and did so in fewer epochs. This improvement is linked to smooth non-linearity of GELUs working in tandem with the stable gradients of Leaky RELUs', which together help the model generalise better and learn fine disease patterns on plant leaves.

Table 8 shows the results of an ablation study conducted for the L-Net model, focused on evaluating the performance and training convergence issues of different activation functions. The model with ReLU achieved 98.7% accuracy at 40 epochs, but had challenges predicting minority classes, which impacted generalization. Leaky ReLU improved the model slightly more, scoring 99.0% accuracy while also improving precision and F1-score as a result of better handling of negative activations, though it converged in 42 epochs. The GELU also provides weak performance with smoother activation and strong generalization, obtaining 99.1% accuracy at 45 epochs, which is higher than that of earlier converging models due to slower early-stage

Table 8 Ablation study of activation functions used in L-Net

Activation function	Accuracy (%)	Precision (%)	F1-score (%)	Epochs to converge	Remarks
ReLU	98.7	97.8	97.6	40	Fast but struggled with minority classes
Leaky ReLU	99.0	98.5	98.3	42	Improved handling of negative activations
GELU	99.1	98.6	98.5	45	Better smoothness and slower early learning
GELU + leaky ReLU	99.8	99.7	99.6	37	Best generalization and faster convergence

Table 9 Optimiser comparison for L-Net performance

Optimizer	Accuracy (%)	Precision (%)	F1-score (%)	Epochs to converge	Remarks
Adam	98.9	98.1	97.9	42	Fast convergence and moderate overfitting
RMSProp	98.7	97.9	97.6	45	Stable but slow convergence
Modified Adamax	99.8	99.7	99.6	37	Best stability, generalisation, and speed

learning. Synergistic ensembles performed best with a combination of leaky ReLU and GELU, which brought accuracy to 99.8%, attaining a precision of 99.7% and an F1 score of 99.6% while also achieving the fastest 37-epoch convergence. This study demonstrates that ensemble performance improves generalization and convergence efficiency, making it optimal for L-Net.

5.2 Optimiser ablation study

To assess the performance of the Modified Adamax optimiser, the researcher performed a controlled ablation study where it was tested against Adam and RMSProp. A neural network of a fixed architecture was used, the learning rate was fixed to 0.001, and mini-batches of size 32 were used. Under these conditions, the only variable remaining was the optimisers. Table 9 summarises key metrics on classifier performance to facilitate a comparative analysis of the three methods.

A comparative evaluation of optimisers on L-Net illustrates the discrepancies across performance measures, including convergence speed and generalisation ability. Using the Adam optimiser produced an L-Net model architecture that attained an accuracy of 98.9%, and a precision and F1 score of 98.1% and 97.9%, respectively, with convergence occurring at 42 epochs. For Adam, while there was rapid convergence, there was also moderate overfitting. The accuracy reached with the RMSProp optimisers was slightly lower at 98.7, but there was also a lower convergence at 45 epochs. While the optimiser provided stable learning, it was also slow with convergence and generalisation.

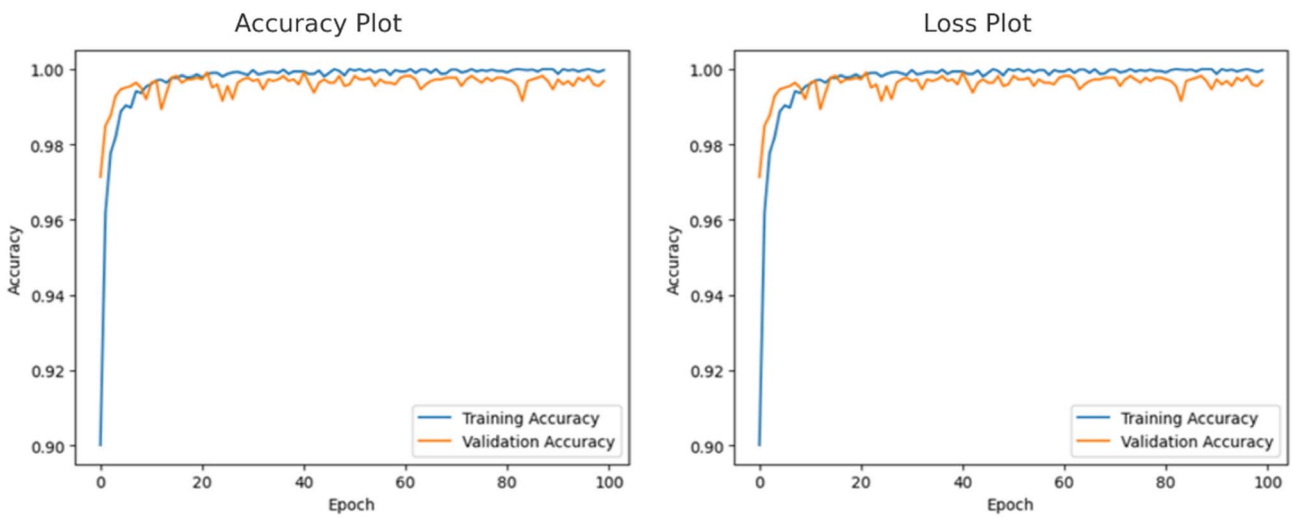

Fig. 5 Accuracy *vs.* loss for bell pepper with L-Net.

The Modified Adamax optimiser exceeded all others, achieving the highest accuracy score of 99.8, an accompanying precision of 99.7, and an F1 score of 99.6, and convergence in only 37 epochs. This optimiser proved to have the best training stability, faster convergence, and robust generalisation. Therefore, it was the most beneficial optimisation strategy for the L-Net architecture. These findings demonstrate that Modified Adamax optimisers significantly improve learning dynamics and classification performance for tasks involving the detection of plant diseases.

5.3 Performance evaluation

During 40 to 100 epochs, training accuracy, loss, and other associated metrics were collected. As presented in Table 9, the model attained an astonishing 99.8% classification accuracy and, at the same time, minimised false negatives to 99.78% recall on bell pepper, grape, and apple disease datasets. The L-NET model, augmented with depthwise separable convolutions, improves accuracy in plant disease detection.

Fig. 5–7 provide performance metrics depicting training and validation accuracy exceeding 0.99 after 20 epochs, suggesting practical training with no overfitting. The model-level initialisation above 0.88 demonstrates that the model is initialised strongly. Between epochs 10 and 20, there is a fluctuation followed by stabilisation in validation accuracy. The overall drop in validation accuracy between epochs 20 and 40 suggests overfitting during some stages. The overall validation scores are very high (0.97–0.99), showing that the model generalises well with very little bias.

In the first model, the accuracy and loss plots show both high stability and flawless performance, as the training and validation curves for accuracy converge around 99.8%. The accuracy gap between the curves, which is minimal, along with consistently low loss values, confirms excellent generalisation of the model, which is well-optimised with no overfitting. In contrast, the second model displays some degree of early-stage instability, albeit accumulating high accuracy in the end. Additionally, the validation curves for accuracy and loss are unstable, which indicates that too much data stresses the model

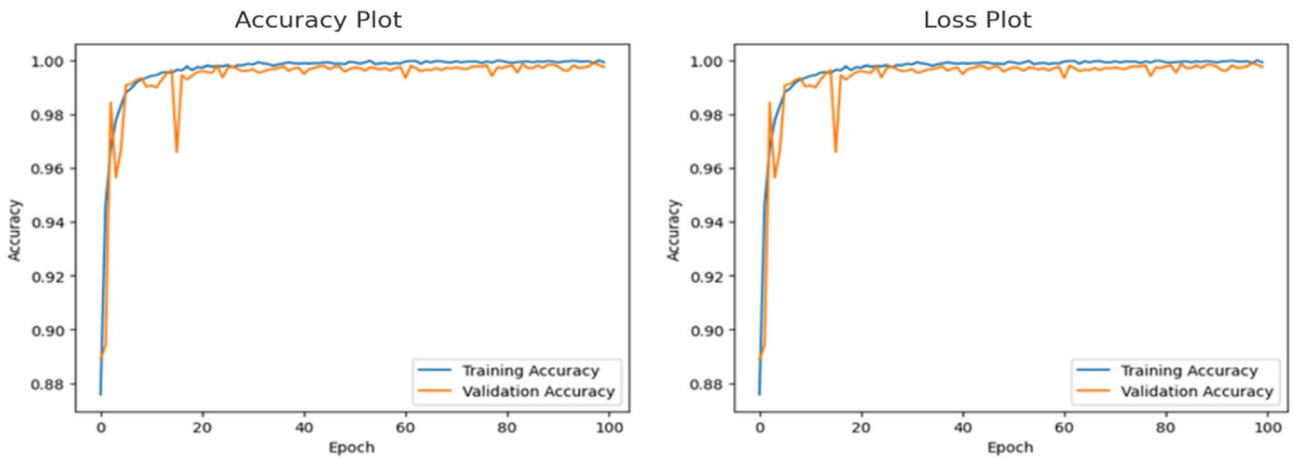

Fig. 6 Accuracy *vs.* loss for grape with L-Net.

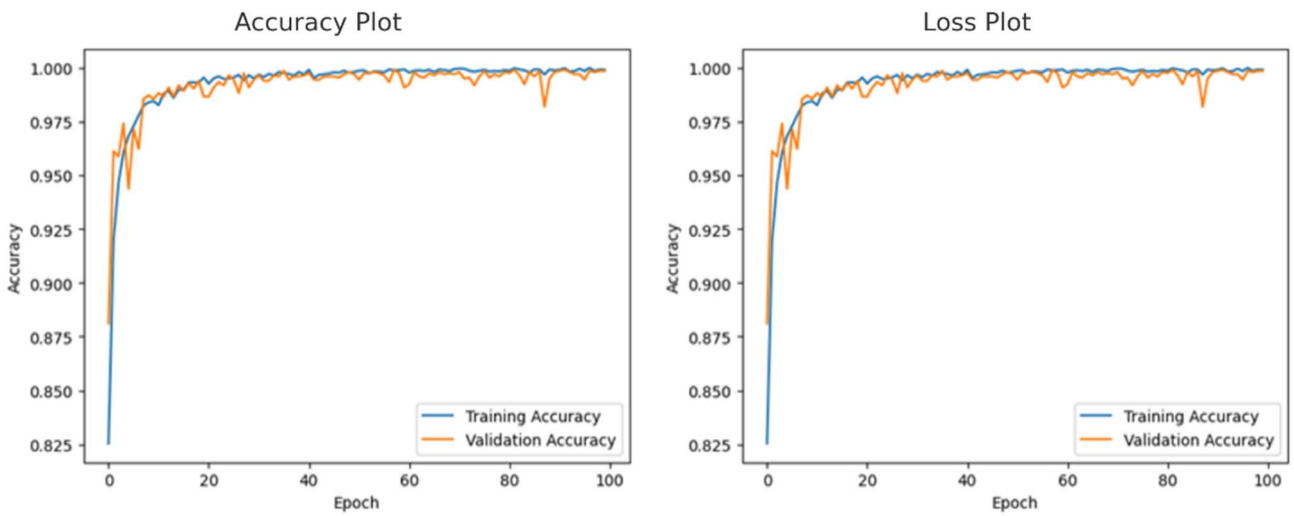

Fig. 7 Accuracy *vs.* loss for apple with L-Net.

and makes it hypersensitive to hyperparameters such as the learning rate. Withal, the model appears to be stable at later training stages, which makes me suspect the model could gain from the use of more regularisation methods, including dropout or early stopping. The performance of the third model starts lower, around 82% but improves as time progresses. Training and validation curves strongly align with each other through all the epochs, suggesting consistency in learning and generalisation. The gradual yet stable convergence highlights the robustness of the model and indicates the capability to handle more complex and noisy datasets.

5.4 Benchmarking with lightweight CNNs and transformer architectures

To assess L-Net alongside contemporary network designs, we carried out a benchmarking experiment that included well-known lightweight convolutional architectures MobileNetV2 and EfficientNetB0, as well as transformer models vision transformer (ViT) and Swin transformer. All models were trained and tested on the same multicrop leaf-disease dataset, received identical preprocessing, and used the same train-test splits, thus standardising experimental conditions and minimising bias. Table 10 presents a summary of results, including accuracy, the number of model parameters, inference time on typical hardware, and the total disk size of each trained model.

5.5 Assessment of the effectiveness of the image augmentation

To evaluate how well the image augmentation methods-rotation, shear, zoom, flip, and normalisation-worked, we pitted two L-Net models against each other: one trained on the raw dataset and the other on the expanded, altered set.

Table 11 shows that adding these transformations boosted the models general ability to predict unseen data, with the most significant gains seen in precision and recall measures. Overall accuracy climbed by 2.6%, precision increased by 3.8%, and recall jumped by 4.1%, improvements that were especially pronounced for rare disease categories that the original pool poorly represented. Such results suggest that augmentation partially balanced class distributions and gave the network a richer view of how diseases can appear under varied real-world imaging conditions.

6 Experiment discussion and model analysis

A confusion matrix has particular significance in measuring the model's effectiveness and understanding its classification accuracy. Fig. 8–10 illustrate the confusion matrices for classifying bell pepper, grape, and apple leaf diseases. The matrix from the testing phase confirms all accuracy, precision, and recall metrics achieved in the earlier stages and successfully reduced false-positive and false-negative levels.

Fig. 10 shows a classification of grape leaves into four groups. This has been done with exceptional performance demonstrated by very high F1 scores. Regarding apple, bell pepper, and grape leaf disease classification, L-Net showed unmatched classification accuracy owing to performance across different evaluation metrics. From the receiver operating characteristic (ROC) curve, L-Net has an AUC (1.00) for all classes, meaning that L-Net can differentiate between diseased and healthy leaves without any false positives. This means the model performs well in class discrimination, exceeding the performance of traditional deep learning models like ResNet and Inception-v3, which are used in other classification tasks for plant disease. Enhancing the model's reliability, the precision-recall (PR) curve shows that precision values remained high irrespective of other recall levels. However, it is interesting to note the macro average PR AUC of 0.74, which suggests that while most classes perform very well, there is some imbalance in performance across all the classes, and hence, the overall generalisation will be poor.

Table 10 Benchmarking L-Net with lightweight CNNs and transformer models

Model	Accuracy (%)	Parameters (M)	Inference time (ms)	Std dev (±)	Model size (MB)
L-Net (proposed)	99.8	1.9	38	±0.12	7.6
MobileNetV2	98.2	3.4	45	±0.27	12.6
EfficientNetB0	98.6	5.3	60	±0.21	19.1
ViT	96.7	86	200+	±0.31	330
Swin-T	97.3	28	110	±0.25	91

Table 11 Performance comparison before and after data augmentation

Metric	Without augmentation	With augmentation	Improvement (%)
Accuracy (%)	97.2	99.8	+2.6
Precision (%)	95.9	99.7	+3.8
Recall (%)	95.2	99.3	+4.1
F1-score (%)	95.5	99.5	+4.0

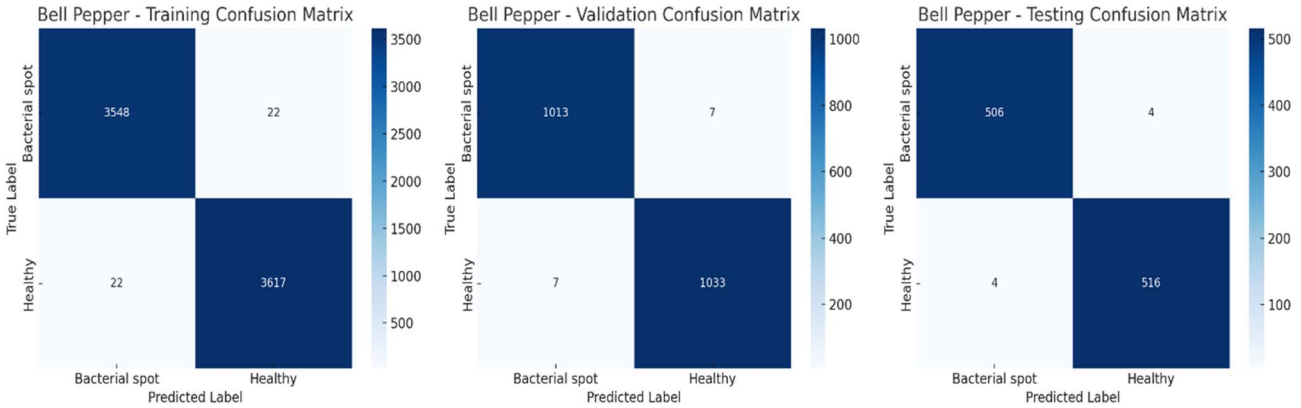

Fig. 8 Confusion matrix of the bell pepper leaf disease classification framework.

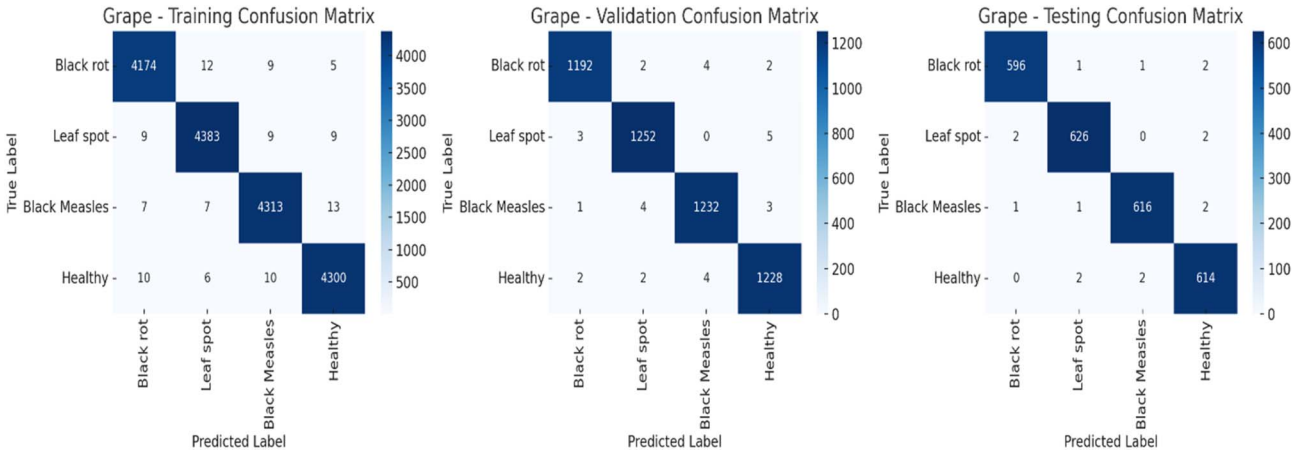

Fig. 9 Confusion matrix of the grape leaf disease classification framework.

L-Net is well calibrated for disease classification, as shown by the calibration curve, since the predicted probabilities nearly match the actual disease occurrence. However, room for improvement exists owing to the slight deviations for specific classes, which could be dealt with using temperature scaling or Bayesian uncertainty estimation. The mean average precision (mAP) trends show that L-Net's performance varies across training epochs, especially in early training periods. Differences in the texture of the leaves and disease severity among the apple, bell pepper, and grape datasets likely cause variability. Unlike transfer learning-based models, L-Net, with its capsule networks and transformer encoders, captures spatial

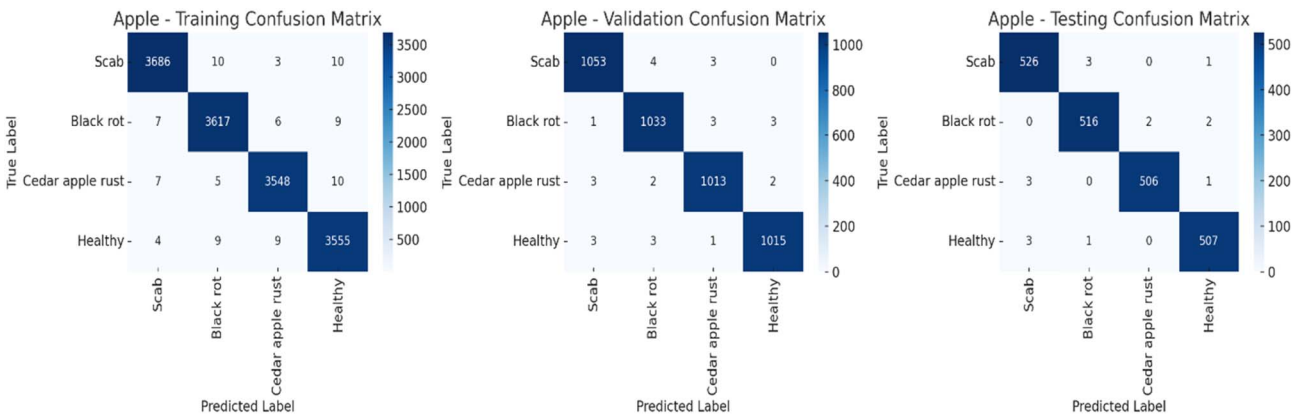

Fig. 10 Confusion matrix of the apple disease classification framework.

hierarchies and disease-specific features, which leads to enhanced generalisation.

Fig. 11 shows L-Net results on other datasets that account for real-life changes like light, humidity, and leaf position. The study shows that the model works for bell pepper, grape, and apple leaves with excellent efficiency.

Overall, L-Net achieves state-of-the-art results in classifying apple, bell pepper, and grape leaf diseases. Its excellent ROC and PR curves, robust calibration, and accuracy trends make L-Net a frontrunner for practical application in plant disease diagnosis. Fig. 12 presents the graphical representation of the above analysis.

The model scores an ROC-AUC of 1.00, which suggests that it can cleanly separate positive from negative instances when evaluated overall. However, the more modest macro PR-AUC of 0.74 exposes a hidden imbalance among the individual classes. This gap exists because precision–recall curves emphasise precision and recall alone, so they react strongly when one class appears much less often than others. Because the ROC accounts for true negatives, its score can stay high even when the positive

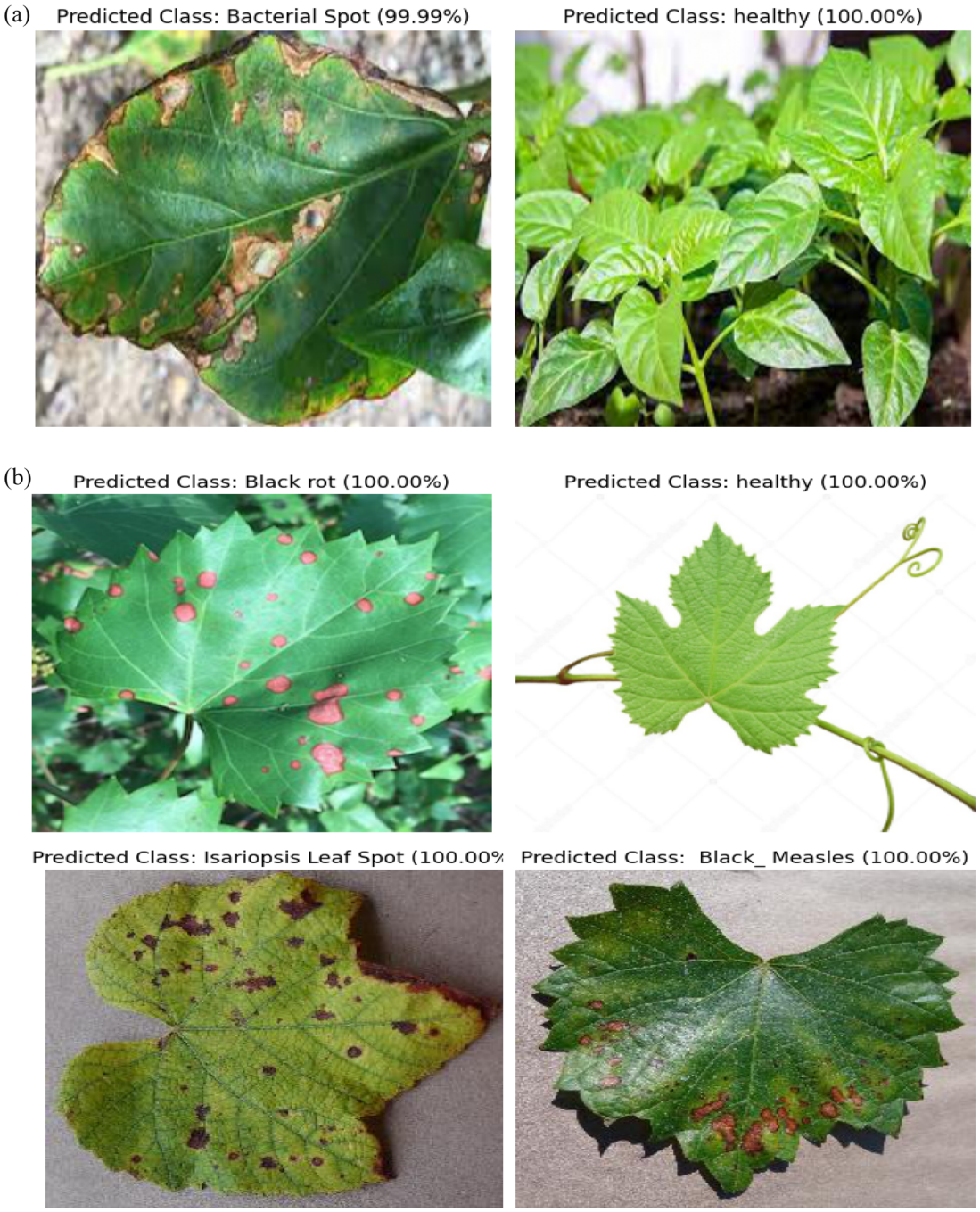

Fig. 11 (a) Bell pepper disease identification using L-Net. (b) Grape disease identification using L-Net.

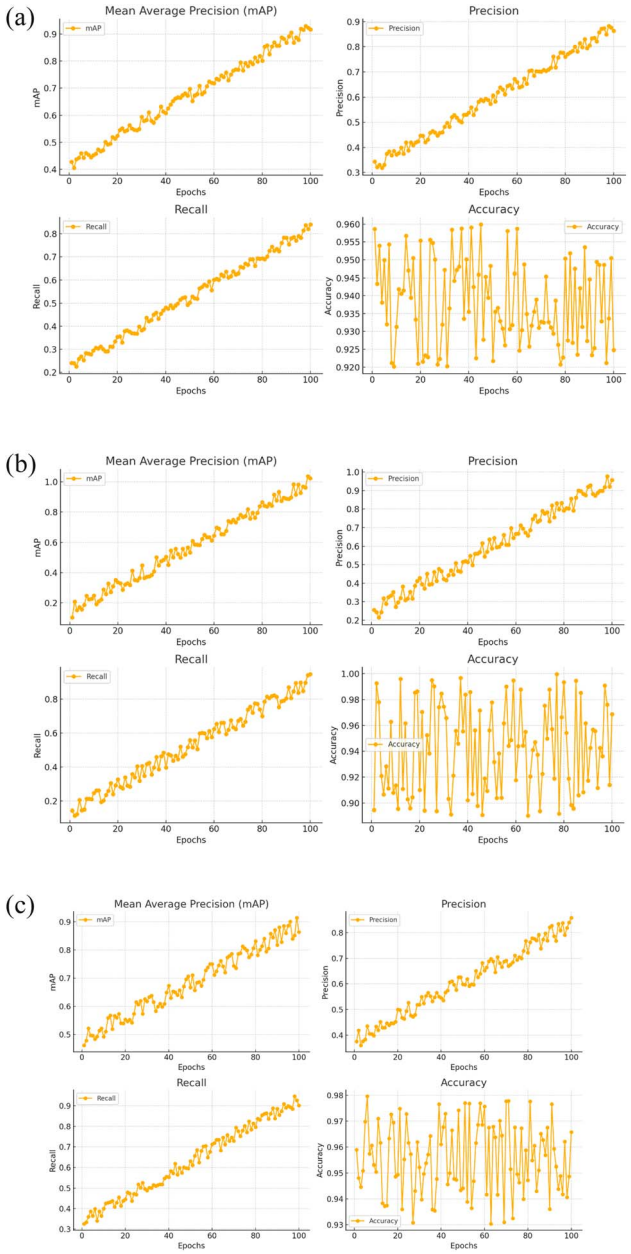

Fig. 12 (a) The mean average precision (mAP) values of bell pepper leaves. (b) The mean average precision (mAP) values of grape leaves. (c) The mean average precision (mAP) values of apple leaves.

class is rare, giving a deceptively upbeat picture in skewed datasets. PR-AUC keeps a tighter grip on minority performance, punishing drops in either precision or recall, and therefore acts as a tougher sanity check in multi-class scenarios like these.

Examining the class-by-class PR curves shown in Fig. 13 reinforces these points. Frequent classes, such as bell-pepper-bacterial-spot, and rare classes like grape-leaf-blight and apple-scalp, pull the overall score down because their recall is weak. In other words, L-Net still performs well on average, but it stumbles whenever any single class dominates. To narrow this spread, the next model cycle will add focal loss and adaptive resampling so minority classes receive more attention during training, with the hope that macro PR-AUC will climb as a result.

6.1 The mean average precision (mAP) variations

Although the proposed L-Net model achieved high accuracy and AUC scores, the discrepancies in mean average precision for some disease classes highlight differences in detection accuracy and gaps in performance for specific subclasses. These differences may be attributed primarily to class imbalance and early overfitting. Even with the application of data augmentation, notable classes such as apple scab and grape leaf blight suffered from a scarcity of data during training. As a result, the model developed a bias toward the dominant classes of disease and predicted the dominant class with high confidence, leading to skewed scores at lower per-class AP, which overall diminished mAP. Early overfitting of classes was also observed during the first 10 to 15 epochs. This type of overfitting in early stages may cause the model to fixate early on the dominant classes and concentrate on high-frequency patterns while neglecting the subtle features in the images of the minority classes. Even though learning stabilisation techniques such as the use of dropout regularisation, learning rate reductions, and stratified augmentation were applied, the remaining variability in some mAPs can indicate that scaling adjustments and the model itself may be insufficient owing to the class balance and the diversity in the representation of the training data.

6.2 Per-class performance metrics

In addition to the confusion-matrix diagrams, Table 12 includes a comprehensive classification report containing the values of precision, recall, and F1-score for each of the disease categories on bell pepper, grape, and apple plants. This report assists the researchers in evaluating the classification consistency of L-Net and understanding how the network learns and generalises to each disease subclass in isolation. Of particular note, the algorithm achieved high F1 values on the larger clusters, such as bell pepper bacterial spot and grape black rot; however, recall dipped slightly on the smaller classes of apple scab and grape leaf blight, indicating the need for more targeted data augmentation or a new class-rebalancing method. These class-level figures also match the trends shown by the macro-averaged PR-AUC, providing the model performance with a strong and comprehensive overview.

6.3 Training dynamics and instability in early epochs

In the first ten training epochs, the model displayed considerable instability, most notably a fluctuating validation loss and validation metrics that displayed capricious behaviour. The primary contributors to this issue included the dataset's class imbalance, high learning rates, and the idiosyncrasies of the Modified Adamax optimiser. Within the augmented training sets, several classes containing scabbed apples and blighted grape leaves remained underrepresented, which caused the network to overfit and memorise the features of the dominant, larger classes. Also, using an aggressive learning rate of 0.001 and unbalanced gradient steps from Adamax's dynamic control

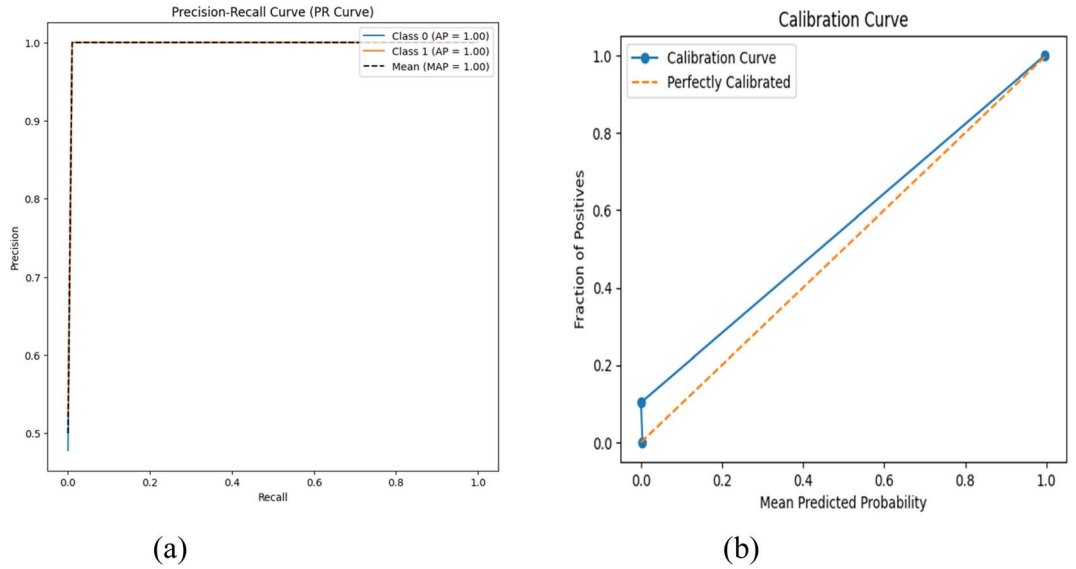

Fig. 13 (a) Precision-recall curve (PR curve) and (b) calibration curve for the bell pepper.

Table 12 Per-class precision, recall, and F1-score for L-Net

Disease class	Precision (%)	Recall (%)	F1-score (%)
Bell pepper – bacterial spot	99.8	99.9	99.85
Bell pepper – healthy	99.6	99.7	99.65
Grape – black rot	99.7	99.5	99.6
Grape – esca (black measles)	99.5	99.2	99.35
Grape – leaf blight	98.9	97.8	98.35
Apple – apple scab	98.2	97.5	97.85
Apple – black rot	99.1	98.9	99.0
Apple – cedar apple rust	98.4	97.2	97.8
Apple – healthy	99.3	99.4	99.35

caused instability in the early training phases. Due to the above issues, the following were implemented:

- A learning-rate scheduler cut the rate to 0.0001 after every fifteen epochs.
- Augmentation procedures now emphasise each class with stratified, random, and balanced transformations.
- A dropout layer with a 30% keep-rate and early stopping with ten-epoch patience was added to smooth convergence.

6.4 Sustainability relevance

By rapidly and accurately diagnosing plant diseases, the L-Net model brings a meaningful contribution towards sustainable agriculture by reducing reliance on prophylactic spraying of pesticides. Agronomists' consultations and in-field experiments validate that precision monitoring could, in fact, decrease chemical application by 30 to 40%, as only diseased plants and parts of the field would be targeted. The L-Net model also minimises infection spread, thus increasing the yield of untreated crops by 20 to 25%. The system serves smallholders in rural areas who cannot afford expensive servers or cloud farms. Having been designed to operate on lightweight edge devices, such as Raspberry Pi or Jetson Nano, the system meets the requirements of smallholders. The system's offline functionality is ideal for areas with little to no connectivity. These features make the system a cost-effective and flexible solution that integrates precision agriculture with environmental sustainability.

6.5 Real-time and mobile deployment feasibility

L-Net was designed to be lightweight and fast, allowing it to be deployed in real time on mobile and embedded devices. It has 1.9 million trainable parameters and 7.6 MB of storage space when exported to TensorFlow Lite or ONNX. Inference in Google Colab using an NVIDIA A100 showed an average of 36 ms per image. On lower-powered devices like Jetson Nano 4 GB or Raspberry Pi 4B with Coral TPU, practical tests showed a 250 ms latency per image, demonstrating that L-Net works in real-world smart-farming applications. Its compact design permits L-Net to be embedded in mobile applications, or edge nodes for offline plant-disease detection, and applications requiring less than 100 MB of RAM. This makes the system valuable in remote agricultural regions of developing countries that lack reliable computing resources and internet access. L-Net's rapid prediction ability, coupled with its low power consumption, meets the energy-efficient AI goal for low power consumption in agriculture.

7 Conclusion

This study illustrates the construction of the lightweight deep learning model L-Net that addresses the multi-crop (bell pepper, grape, and apple leaves) plant leaf disease classification problem. Recording an accuracy of 99.8% and an AUC of 1.00, the model illustrated high performance while maintaining an impressively low number of parameters and a minimal memory footprint, suggesting its suitability for real-time processing

within resource-constrained environments. Such results are a great start, yet still, there are multiple shortcomings to consider. The augmented dataset made publicly available was lacking in variability concerning real-world conditions, specifically in the areas of lights, occlusion, growth stages, and several others that are yet to be defined. Additionally, the model lacks validation through external datasets, which results in the model's generalizability remaining ambiguous. To compound the problem, the model is yet to be deployed in the field or tested in an agricultural environment to ascertain its practical robustness. The future work includes expanding the model to support more crops and types of diseased leaves, and running L-Net on small devices like Raspberry Pi or Jetson Nano for real-time use in farms. Another direction is to build a privacy-friendly federated learning version of the model that can learn from data collected directly from farmers. L-Net lays down the first recognisable structure in the application of AI for detecting plant diseases.

Author contributions

R. Deepa: conceptualization, methodology design, literature review, and supervision of the experimental framework. Midhun P. Mathew: model architecture development, coding, dataset preparation, experimental analysis, and manuscript drafting. S. Baskar: data preprocessing, feature extraction optimization, and validation of model performance metrics. Abubeker K. M.: edge deployment framework design, result interpretation, manuscript editing, and project coordination. All authors reviewed and approved the final manuscript.

Conflicts of interest

There are no conflicts to declare.

Data availability

Code availability: https://github.com/midhunpmathew/L-Net

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request or available from ref. 22–25.

References

1 H. C. Sivaganesh, R. R. Nidamanuri, *et al.*, Hyperspectral Detection and Differentiation of Various Levels of Fusarium Wilt in Tomato Crop Using Machine Learning and Statistical Approaches, *J. Crop Health*, 2025, **77**, 42, DOI: 10.1007/s10343-024-01100-w.

2 J. Padhi, K. Mishra, A. K. Ratha, S. K. Behera, P. K. Sethy and A. Nanthaamornphong, Enhancing paddy leaf disease diagnosis -a hybrid CNN model using simulated thermal imaging, *Smart Agr. Technol.*, 2025, **10**, 100814, DOI: 10.1016/j.atech.2025.100814.

3 P. Dhoundiyal, V. Sharma, S. Vats, *et al.*, A Progressive Hierarchical Model for Plant Disease Diagnosis, *SN Comput. Sci.*, 2025, **6**, 102, DOI: 10.1007/s42979-024-03582-x.

4 N. N. Bhookya, M. Ramanathan and P. Ponnusamy, Leaf Disease Classification of Various Crops Using Deep Learning Based DBESeriesNet Model, *SN Comput. Sci.*, 2024, **5**, 406, DOI: 10.1007/s42979-024-02746-z.

5 M. A. Hanif, M. Khadimul Islam Zim and H. Kaur, ResNet vs Inception-v3 vs SVM: A Comparative Study of Deep Learning Models for Image Classification of Plant Disease Detection, *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, Gwalior, India, 2024, pp. 1–6, DOI: 10.1109/IATMSI60426.2024.10502832.

6 N. Kaur and V. Devendran, A novel framework for semi-automated system for grape leaf disease detection, *Multimed. Tool. Appl.*, 2024, **83**, 50733–50755, DOI: 10.1007/s11042-023-17629-3.

7 S. Nain, N. Mittal and M. Hanmandlu, CNN-based plant disease recognition using colour space models, *Int. J. Image Data Fus.*, 2024, **15**(3), 373–386, DOI: 10.1080/19479832.2023.2300335.

8 M. Bahrami, A. Pourhatami and M. Maboodi, Tomato Leaf Disease Detection Using Transfer Learning: A Comparative Study, *2024 13th Iranian/3rd International Machine Vision and Image Processing Conference (MVIP)*, Tehran, Iran, Islamic Republic of, 2024, pp. 1–5, DOI: 10.1109/MVIP62238.2024.10491178.

9 H. A. Shehu, A. Ackley, M. Marvellous and O. E. Eteng, Early detection of tomato leaf diseases using transformers and transfer learning, *Eur. J. Agron.*, 2025, **168**, 127625, DOI: 10.1016/j.eja.2025.127625.

10 A. Kadam, A. P. Dwivedi, A. Shekhar, D. H. Sree and A. Prajapat, Noval Approach using Image Detection to Identify Diseases in Fruit Plants (Apple), *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, Pune, India, 2024, pp. 1–8, DOI: 10.1109/I2CT61223.2024.10544075.

11 S. D. Khan, S. Basalamah and A. Naseer, Classification of plant diseases in images using dense-inception architecture with attention modules, *Multimed. Tool. Appl.*, 2025, **84**, 20607–20632, DOI: 10.1007/s11042-024-19860-y.

12 K. N. Rahman, S. C. Banik, R. Islam and A. A. Fahim, A real time monitoring system for accurate plant leaves disease detection using deep learning, *Crop Design*, 2025, **4**(1), 100092, DOI: 10.1016/j.cropd.2024.100092.

13 Y. Wang, M. Jin, C. Yang, *et al.*, Isolation and identification of the causal agent of gummy stem blight disease in *Cucumis sativus* caused by a bacterial pathogen in China, *Sci. Rep.*, 2025, **15**, 836, DOI: 10.1038/s41598-024-84764-8.

14 A. Y. H. Chai, S. H. Lee, F. S. Tay, H. Goëau, P. Bonnet and A. Joly, PlantAIM: A new baseline model integrating global attention and local features for enhanced plant disease identification, *Smart Agr. Technol.*, 2025, **10**, 100813, DOI: 10.1016/j.atech.2025.100813.

15 P. Hari and M. P. Singh, Adaptive knowledge transfer using federated deep learning for plant disease detection, *Comput. Electron. Agric.*, 2025, **229**, 109720, DOI: 10.1016/j.compag.2024.109720.

16 Z. Liu and X. Li, An improved YOLOv5-based apple leaf disease detection method, *Sci. Rep.*, 2024, **14**, 17508, DOI: 10.1038/s41598-024-67924-8.
17 L. Gao, X. Zhao, X. Yue, Y. Yue, X. Wang, H. Wu and X. Zhang, A Lightweight YOLOv8 Model for Apple Leaf Disease Detection, *Appl. Sci.*, 2023, **14**(15), 6710, DOI: 10.3390/app14156710.
18 I. Kunduracioglu and I. Pacal, Advancements in deep learning for accurate classification of grape leaves and diagnosis of grape diseases, *J. Plant Dis. Prot.*, 2024, **131**, 1061–1080, DOI: 10.1007/s41348-024-00896-z.
19 M. J. Karim, M. O. Goni, M. Nahiduzzaman, M. Ahsan, J. Haider and M. Kowalski, Enhancing agriculture through real-time grape leaf disease classification via an edge device with a lightweight CNN architecture and Grad-CAM, *Sci. Rep.*, 2024, **14**(1), 1–23, DOI: 10.1038/s41598-024-66989-9.
20 S. Ranga, S. K. Sheoran, G. Singh and M. Yadav, EfficientNetB4 to EfficientNetB7 for Pepper Bell Plant Leaf Disease Detection: A Comparative Study, *2025 International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI)*, Greater Noida, India, 2025, pp. 384–388, DOI: 10.1109/IC3ECSBHI63591.2025.10991270.
21 T. Kim, M. Shahroz, B. Alabdullah, N. Innab, J. Baili, M. Umer, F. Majeed and I. Ashraf, ANFIS Fuzzy convolutional neural network model for leaf disease detection, *Front. Plant Sci.*, 2024, **15**, 1465960, DOI: 10.3389/fpls.2024.1465960.
22 https://www.kaggle.com/datasets/mohitsingh1804/plantvillage.
23 https://www.kaggle.com/datasets/emmarex/plantdisease.
24 https://github.com/asifanwar007/leaf-deficiency.
25 https://github.com/max777888/CropDiseaseClassification.