Journal of Edge Computing, 2025, Vol. **4**, Iss. 2, pp. 234–262 https://doi.org/10.55056/jec.905

Comparative and edge-hybrid modeling of EfficientNetV2 and MobileNetV2 for multi-class crop disease classification with statistical validation

Thomas Kinyanjui Njoroge[1], Rachael Kibuku[2] and Kevin Mugoye[2]

[1] *Karatina University, School of Pure and Applied Sciences, P.O. Box 1957–10101, Karatina, Kenya*
[2] *KCA University, School of Technology, P.O. Box 56808–00200, Nairobi, Kenya*

Abstract. Crop disease classification is critical for global food security, yet deploying accurate deep learning models on resource-constrained edge devices remains challenging. This study systematically compares EfficientNetV2 and MobileNetV2 while proposing an edge-optimised hybrid architecture integrating both with vision transformers (ViT). Evaluated on PlantVillage and field-collected images, MobileNetV2 demonstrated superior edge compatibility with 99.0% accuracy, 0.0938 s/image inference speed, minimal resources (30.38 MB size), and statistical superiority (z-test p=0.0071). The hybrid model combines MobileNetV2's texture analysis and EfficientNetV2's multiscale detection through a dual-branch architecture enhanced with SE blocks, ViT (16×16 patches), and attention-guided fusion. It achieved 99.5% test accuracy with real-time performance (0.15 s/image) and 97.97% field accuracy via Android deployment. Statistical validation confirmed robustness: Kruskal-Wallis H=597.40 (p<0.05), near-perfect AUC (0.999998), and minimal confidence variance (0.000010). Ablation studies verified architectural efficacy (98.68% accuracy with SE/gating modules). This work advances precision agriculture through a scalable framework unifying hybrid deep learning with edge-compatible deployment.

Keywords: crop disease classification, deep learning, convolutional neural networks, EfficientNetV2, MobileNetV2

1. Introduction

The global agricultural sector faces significant challenges from crop diseases, which threaten food security and economic stability, necessitating reliable methods for early and accurate disease detection. Karypidis et al. [16] assert that traditional approaches, reliant on expert manual visual inspection, are labour-intensive, error-prone, and impractical for large-scale deployment. In recent years, deep learning (DL) has emerged as a transformative solution, with convolutional neural networks (CNNs) demonstrating exceptional success in automating crop disease classification by analysing leaf images. Among DL architectures, models like MobileNetV2 and EfficientNetV2 have gained prominence for their ability to balance accuracy with computational efficiency – a critical requirement for real-world agricultural applications, particularly in resource-constrained environments. However, Bernardes et al. [4] point out that the proliferation of these models has introduced a new challenge: identifying an optimal architecture that effectively balances accuracy, efficiency, and real-time responsiveness. While MobileNetV2 is lauded for its lightweight design, as demonstrated in the works of Dai et al. [7], making it ideal for mobile and edge devices, EfficientNetV2 is recognised for its superior accuracy and training efficiency through compound scaling. Abdu,

0009-0000-2147-9848 (T. K. Njoroge); 0000-0002-9590-0882 (R. Kibuku)
tnjoroge@karu.ac.ke (T. K. Njoroge); rkibuku@kcau.ac.ke (R. Kibuku); kmugoye@kcau.ac.ke (K. Mugoye)

Mokji and Sheikh [2] demonstrate that existing studies often evaluate these models in isolation or on disparate datasets, resulting in inconsistent conclusions about their comparative performance. Cecaj et al. [5] further emphasised that this issue is exacerbated by the lack of rigorous statistical validation, which is crucial for confirming the reliability of reported results. This ambiguity complicates decision-making for stakeholders seeking deployable solutions.

This study addresses key research gaps using a dual approach: a comparative analysis and a hybrid multi-class crop disease classification model. First, MobileNetV2 and EfficientNetV2 are benchmarked under identical conditions using standardised datasets to ensure fair comparison. Metrics such as accuracy, precision, recall, F1-score, inference time, and computational cost are used to evaluate performance. Second, we propose a novel hybrid model combining MobileNetV2, EfficientNetV2, and Transformers, enhanced with SE blocks, gating mechanisms, and multiscale fusion for efficient feature extraction.

The model was trained on a combined dataset of 76 classes from PlantVillage (https://plantvillage.psu.edu/) and locally collected images. Statistical methods – including McNemar's test, Cohen's kappa, z-test, t-test, and confidence analysis – validated the comparative and hybrid results, quantifying trade-offs between accuracy and efficiency. This multi-dimensional analysis offers practical insights for stakeholders balancing accuracy for diagnostics and efficiency for edge deployment. The paper is structured as follows: section 2 reviews related work, section 3 covers methods, section 4 presents results, and section 5 discusses findings and future directions.

2. Related work

Integrating deep learning into agricultural systems has marked a paradigm shift in crop disease identification, overcoming limitations inherent to traditional and early machine learning approaches. DL's ability to autonomously learn hierarchical features from raw images has proven transformative, particularly with the adoption of CNNs. Early applications of CNNs, as shown in the works of Chen et al. [6] and Shah et al. [31], demonstrated unprecedented accuracy in classifying diseases using curated datasets like PlantVillage. These studies highlighted DL potential to generalise across diverse disease patterns, achieving accuracies exceeding 95% in controlled environments. Subsequent research focused on enhancing robustness to real-world variability, such as varying lighting, occlusions, and background clutter. For instance, Liu et al. [18] employed deeper architectures to classify apple leaf diseases under field conditions. Padshetty and Ambika [24] demonstrated that fine-tuning pre-trained on domain-specific datasets could achieve near-perfect accuracy. These efforts underscored the importance of transfer learning, where models pre-trained on large-scale datasets were adapted to agricultural tasks, mitigating data scarcity in specialised domains.

However, the computational complexity of early DL models posed challenges for deployment in resource-constrained settings. Researchers began exploring strategies to balance accuracy with efficiency, such as network pruning, as shown by Liu et al. [19] and ensemble by Ge et al. [12], and architectural innovations tailored for edge devices. Abdu, Mokji and Sheikh [2] benchmarked lightweight CNNs against traditional models, revealing that even simplified architectures could achieve comparable accuracy with significantly reduced computational overhead. As deep learning adoption expanded, the demand for models optimised for mobile and IoT devices became crucial. This has spurred the development of efficiency-oriented architectures, prioritising parameter reduction and inference speed without sacrificing accuracy. Techniques like depthwise separable convolutions [21] and inverted residual blocks, as in the work of Dhaka et al. [8], became foundational to these designs, enabling real-time processing on devices with limited computational resources. While early lightweight models faced trade-offs between accuracy and speed, iterative refinements gradually bridged this gap, paving the way for their integration into field-deployable tools.

EfficientNetV2 has shown promise in handling the fine-grained features of crop diseases in agricultural applications. For example, Abasi et al. [1] applied EfficientNet-B2 to classify rice leaf diseases, achieving 98.4% accuracy on a dataset with complex background noise, outperforming ResNet-50 and DenseNet-121. Similarly, a study by Sun et al. [32] demonstrated its efficacy in detecting tomato leaf diseases, where EfficientNetV2 achieved 99.70% accuracy. The model's ability to generalise across small and imbalanced datasets – common challenges in agricultural imaging – has been attributed to its progressive learning strategy, which adjusts regularisation dynamically during training. Despite its strengths, EfficientNetV2's larger variants, such as EfficientNetV2-L, remain computationally intensive for edge deployment, requiring trade-offs between accuracy and real-time performance. However, its smaller variants, such as EfficientNetV2-B0, retain the benefits of compound scaling while maintaining a compact footprint, making them viable for IoT-based agricultural systems. These characteristics position EfficientNetV2 as a versatile solution for high-accuracy disease classification, mainly when computational resources are moderately available. MobileNetV2, as shown by Dong et al. [10], represents a significant leap in designing lightweight CNNs tailored for mobile and edge-device applications. Building on the success of MobileNetV1, as in the work of Glegoła, Karpus and Przybyłek [14], popularised depthwise separable convolutions to reduce computational costs, MobileNetV2 has been widely adopted for real-time disease detection due to its deployability on low-power devices. Pineda Medina et al. [27] deployed MobileNetV2 to diagnose potato diseases, achieving 98.7% accuracy. However, its applicability depends on dataset quality, environmental variations, and the model's ability to generalise beyond controlled conditions. Fang, Zhen and Li [11] integrated MobileNetV2 into an edge-computing system for in-field wheat disease identification, reporting a 98.7% accuracy rate under variable lighting and occlusion conditions. Its efficiency is also suitable for federated learning frameworks, where models are trained collaboratively across distributed devices without centralised data storage. However, MobileNetV2's lightweight design entails trade-offs. Comparative studies, such as Saleem et al. [29], found that while MobileNetV2 outperformed VGG16 and ResNet50 in inference speed, it lagged marginally in accuracy on the PlantVillage dataset. This accuracy gap widens in scenarios requiring fine-grained classification, such as distinguishing between visually similar diseases, where deeper models like EfficientNetV2 often excel, as demonstrated by Sun et al. [32]. Nevertheless, MobileNetV2 remains a cornerstone for applications prioritising real-time performance and low resource consumption, particularly in regions with limited computational infrastructure.

Statistical validation is a cornerstone of robust ML research, ensuring that reported performance metrics are reliable, reproducible, and not attributable to random chance. Unlike traditional accuracy-centric evaluations, as shown by Cecaj et al. [5], statistical methods account for variability in data sampling, model initialisation, and hyperparameter tuning, which, according to [23], can lead to overoptimistic or misleading conclusions. In agricultural applications, where model deployability hinges on consistent performance under diverse field conditions, rigorous statistical validation becomes indispensable. A common pitfall in ML studies is reliance on single-trial accuracy scores, which ignore variance across different data splits or training runs. For instance, a model achieving 95% accuracy in one trial might drop to 92% in another due to stochastic factors, such as random weight initialisation or data shuffling, as shown by Dong, Liu and Tham [9] to mitigate this, researchers employ techniques like k-fold crossvalidation, as described by Phinzi, Abriha and Szabó [26] which partitions the dataset into k subsets and iteratively tests the model on each fold, averaging results to reduce bias. Orchi et al. [23] used 10-fold cross-validation to evaluate traditional and deep transfer learning. While this enhances model reliability, repeated training increases computational cost. Ojo and Zahid [22] correctly points out that controlled validation performance may not fully reflect real-world generalizability, where diverse field conditions introduce variability.

Recent advancements synthesise efficient CNN backbones [15], attention-based context modelling [13], and adaptive fusion techniques within modular frameworks. However, a key gap persists in quantifying and comparing the cost-effectiveness of these components in hybrid systems. While models such as Swin Transformer [20, 34] represent milestones in vision architecture, they often treat fusion as a monolithic process, overlooking the granular analysis of individual modules. The evolving literature supports a shift toward fine-grained, cost-aware multiscale fusion strategies vital for deployment in low-resource environments. Evaluating hybrid vision architectures necessitates cautiously balancing computational efficiency and predictive accuracy. As such, recent studies have adopted a range of metrics to systematically quantify this trade-off across different components and deployment environments. Complexity metrics such as FLOPs [35] and parameter counts [17] remain foundational for assessing computational cost. A granular analysis is particularly informative in hybrid models; for instance, adding squeeze-and-excitation (SE) blocks minimises the overall parameter count while enhancing channel-wise feature recalibration. Moreover, latency is increasingly used as a real-world performance indicator, particularly in edge computing scenarios. To this end, latency measurements are typically conducted on resource-constrained platforms such as Raspberry Pi and high-performance servers with GPU acceleration. On the performance side, top-1 classification accuracy remains a standard benchmark, though recent work emphasises relative improvements over absolute scores.

3. Materials and methods

3.1. Model architectures

This section presents the independent architectures of EfficientNetV2, as shown in figure 1, and MobileNetV2 in figure 2, along with the proposed model in figure 3 for multi-class crop disease classification. Each model was trained separately under identical conditions to facilitate a fair performance comparison. In these architectures, MBConv (Mobile Inverted Bottleneck Convolution) layers, such as MBConv v6, represent inverted residual blocks with an expansion ratio of 6. At the same time, Fused MBConv v4 combines the initial 1×1 convolution and the subsequent 3×3 convolution into a single operation to reduce memory access and improve efficiency. These design choices enhance both computational performance and representational capacity in resource-constrained environments.

MobileNetV2 achieves high accuracy on image classification tasks while optimised for real-time applications, making it well-suited for low-power devices, edge AI, and mobile vision applications.

3.2. Proposed model architecture

The hybrid model features a dual-path CNN backbone that leverages MobileNetV2 to efficiently extract fine texture details suited to edge computing alongside EfficientNetV2, which captures multiscale hierarchical features critical for assessing disease severity. Channel-wise recalibration modules refine the feature representations and amplify important diagnostic cues while reducing irrelevant noise. Complementing the CNN branches, a vision transformer processes image patches of size 16×16 with positional encoding to capture global spatial relationships, enabling the model to

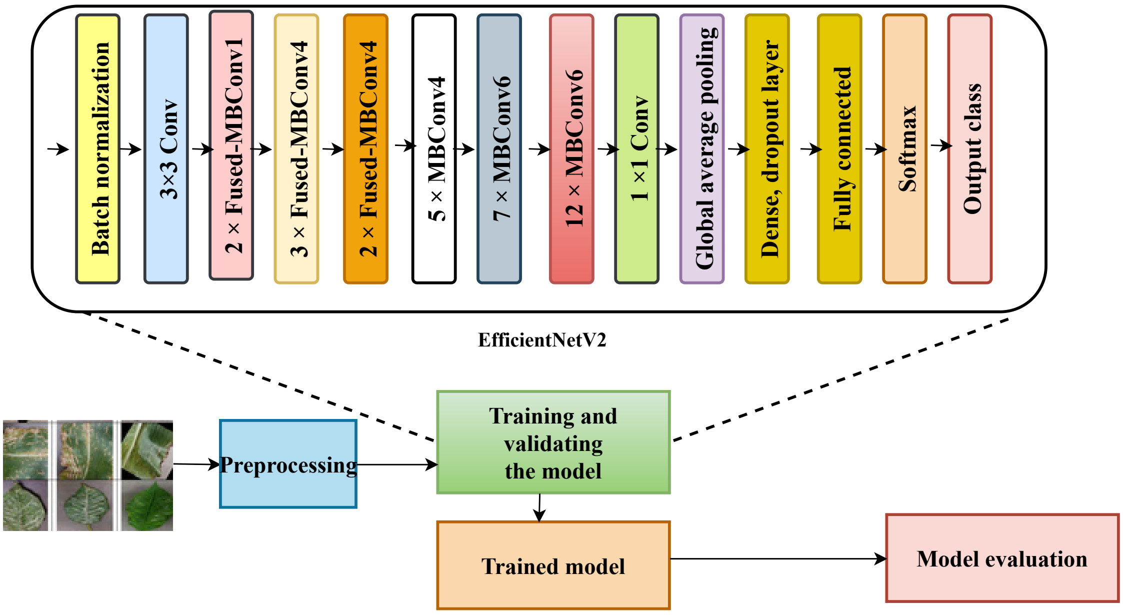


Figure 1: EfficientNetV2 architecture.

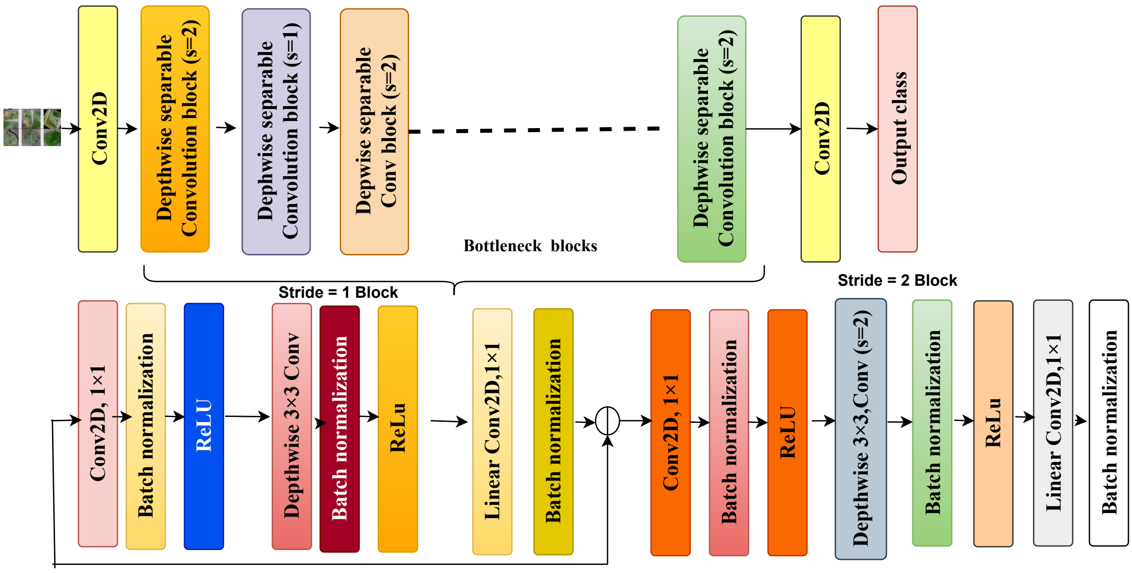


Figure 2: MobileNetV2 architecture.

understand lesion distribution beyond the localised focus of CNNs. The outputs from CNNs and ViT are combined through an attention-based fusion mechanism that uses gated spatial and channel attention to emphasise disease-relevant regions. The model also incorporates multiscale feature aggregation by merging convolutional outputs from kernels of different sizes (3×3, 5×5, and 7×7) hierarchically, enhancing robustness against lesion size and image scale variations. The training process applies batch normalisation and dropout to prevent overfitting, with a softmax classifier designed to distinguish among 76 classes representing various crop diseases and healthy states, ensuring a balanced trade-off between accuracy and generalisation on agricultural datasets.

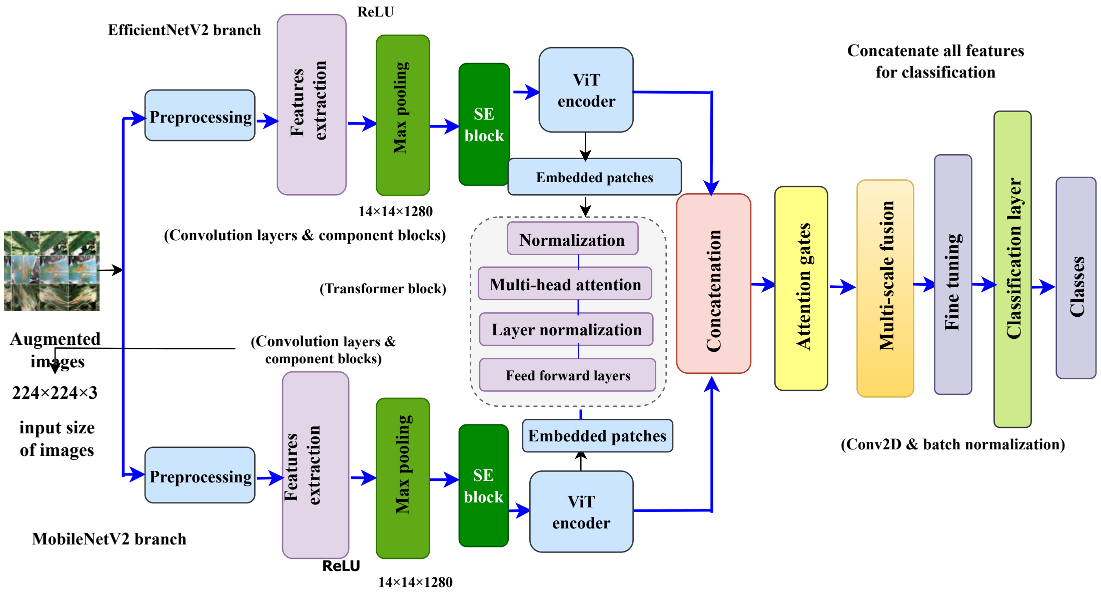


Figure 3: Proposed model architecture.

3.3. Dataset preparation and preprocessing

This section presents the dataset preparation and preprocessing techniques employed in this study. It outlines the steps followed, data sources, preprocessing steps, and strategies to enhance model performance and ensure robust feature extraction.

Step 1: Loading and organising data.

The dataset was loaded from a directory structure where images were organised in subfolders based on their class labels. TensorFlow's image dataset directory function was used to load the data with shuffling, batching, and resizing.

Step 2: Data augmentation.

Off-the-fly data augmentation was applied during training to enhance model generalisation. Table 1 provides details of the augmentation techniques used:

Table 1
Data augmentation.

Transformation type	Range/details
Rotation	0°, 90°, 180°, or 270°
Flipping	Horizontal flip, vertical flip
Brightness adjustment	Between 0.7 (dark) and 1.3 (bright)
Zoom	Resizing and cropping to 224×224 pixels

Step 3: Normalisation of pixel values.

After loading the dataset, pixel values were normalised to a range of [0,1] to enhance model convergence during training. The normalisation of pixel values was mathematically expressed as follows:

$$Normalized\_pixel = \frac{pixel\_value}{255.0} \tag{1}$$

Step 4: Stratified sampling for data splitting.

A stratified sampling technique was employed to ensure that the distribution of classes within the training and validation datasets remained consistent with the original dataset. The number of samples designated for the training set based on stratified sampling was calculated as follows:

$$Train\_size = \frac{number\_of\_samples\_in\_class}{total\_samples} \times total\_samples \times (1 - test\_size) \tag{2}$$

We selected an image size of 224×224×3 because it aligns with the standard input dimensions used in widely adopted deep learning architectures such as VGG, ResNet, MobileNet, and EfficientNet. The images were processed in RGB format, where the three channels (red, green, and blue) capture essential colour variations such as leaf texture, lesions, and discolouration, which are critical for accurate disease detection. This resolution ensures compatibility with pre-trained models during transfer learning, balances computational efficiency, and retains sufficient visual detail for effective feature extraction.

For dataset partitioning, train_size represented the number of samples allocated to the training set, while total samples denoted the full dataset size. A test size of 0.2 was applied to create an 80–20 split between training and validation, a widely accepted practice to ensure robust model generalisation. Additionally, a random state of 42 was used to guarantee reproducibility of the split across multiple runs.

Step 5: Categorical labelling and one-hot encoding.

The dataset, structured into class-specific subdirectories, employed categorical labelling with one-hot encoding for class identification. This ensured that the dual-input model architecture received the same preprocessed image for both branches, maintaining consistency during training.

3.4. Features extraction

The feature extraction process followed a structured approach to ensure feature representation for real-time crop disease detection, as shown in the following steps:

Step 1: Input image preprocessing.

The input images were initially preprocessed to ensure compatibility with the MobileNetV2 and EfficientNetV2 architectures. Each image was resized to a fixed dimension, denoted as H and W×3, where H and W represent the height and width of the input image. 3 corresponds to the RGB colour channels. These preprocessed images were then represented as x_{input} and x_{input2} for MobileNetV2 and EfficientNetV2, respectively.

Step 2: Feature extraction using pre-trained models.

The input images were fed into MobileNetV2 and EfficientNetV2, pre-trained on the ImageNet dataset, to extract meaningful feature representations and were defined as:

$$f_{mobile} = F_{MobileNetV2}(x_{input1}) \quad and \quad f_{efficient} = F_{EfficientNetV2}(x_{input2}) \tag{3}$$

Step 3: Channel attention mechanism via SE network

An SE attention mechanism was applied to enhance the representational power of the extracted features. The SE block first computed the global average pooling for each feature map channel as follows:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X(i, j, c) \tag{4}$$

The pooled values transformed the fully connected layers and a sigmoid activation function to generate channel-wise attention weights, and were expressed as:

$$s = \sigma(W_2 \cdot ReLU(W_1 \cdot z)) \tag{5}$$

These attention weights were then applied to the feature maps to enhance informative channels while suppressing less significant ones, yielding refined feature representations as shown:

$$F^{SE}_{MobileNet} = SE(F_{MobileNet}) \quad and \quad F^{SE}_{EfficientNet} = SE(F_{EfficientNet}) \tag{6}$$

Step 4: Spatial attention for enhancing feature localisation.

After channel refinement, a spatial attention mechanism was incorporated to further focus on discriminative regions within the image. The process involved computing a spatial descriptor through global average pooling, followed by two transformation layers that produce spatial attention scores, given as:

$$d_1 = \sigma(W_1 \cdot avg\_pool + b_1) \quad \text{and} \quad d_2 = \sigma(W_2 \cdot d_1 + b_2) \tag{7}$$

The feature maps were then modulated using these scores, leading to spatially enhanced feature representations, denoted as $F' = F \times d_2$.

Step 5: Multiscale feature fusion.

Inspired by Inception-style architectures, the multiscale fusion module synthesised diverse spatial features for a holistic representation. The outputs were concatenated along the channel axis, ensuring a unified multiscale representation as shown:

$$F_{fused} = concat(F'_{MobileNet}, F'_{EfficientNet}) \tag{8}$$

Step 6: Feature normalisation and dimension reduction.

To stabilise training and enhance generalisation, batch normalisation was applied to the fused feature representation and was calculated as follows:

$$dense_{output} = ReLU(W_d \cdot F_{fused} + b_d) \tag{9}$$

Step 7: Classification using softmax activation.

The final stage involved passing the refined features through a classification layer equipped with a softmax activation function. This function computed the probability distribution over the output classes and was calculated as follows:

$$P(y) = softmax(W_s \cdot dense_{output} + b_s) \tag{10}$$

The class with the highest probability was selected as the final prediction, determining the specific disease label for the given input image. The trained model was then evaluated using accuracy, precision, recall, and F1-score, and these were calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{14}$$

where TP is a true positive, TN is a true negative, FP is a false positive, and FN is a false negative. The confusion matrix was defined as:

$$CM = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \tag{15}$$

The receiver operating characteristic (ROC) curves, the true positive rate (TPR) against the false positive rate (FPR) was calculated as follows:

$$TPR = \frac{TP}{TP+FN}, \quad FPR = \frac{FP}{FP+TN} \tag{16}$$

The area under the curve (AUC) measured classification performance and were calculated as follows:

$$AUC = \int\limits_{0}^{1} TPR(FPR)\, d(FPR) \tag{17}$$

3.5. Dataset description

This study created a combined dataset in table 2 by integrating the Kaggle dataset (38 classes, 60,343 images) [28] with the FieldPlant dataset (25,775 images) from Central Kenya. The FieldPlant dataset accounted for seasonal variations, emphasising fungal and bacterial diseases during April, May, October, and November while prioritising viral infections in June, July, and December. A standardised collection process ensured diverse lighting conditions and angles, enhancing generalisation. Images were classified and annotated by an agricultural expert, and data augmentation techniques were applied to address class imbalances by generating additional samples. The dataset was subsequently divided into training, validation, and test subsets. The training set was used for model learning, the validation set for hyperparameter tuning and early stopping, and the independent test set for final evaluation.

The dataset's diversity and class distribution are presented in table 3. The initial batch comprises 47,627 images covering crop types such as apple, grape, maise, tomato, and potato, with a separate training and validation set for each class. The dataset included multiple disease classes per crop to ensure comprehensive representation, supporting multi-class classification tasks. Importantly, class distribution was carefully analysed before training. Although some variation in sample counts existed across classes, stratified sampling was employed during dataset splitting to preserve the proportional representation of each class in both training and validation sets. This approach mitigated potential bias and helped the model learn effectively from both majority and minority classes, thus supporting accurate and generalisable real-time plant disease detection.

Table 4 details the second batch, spanning classes from tomatoes to beans, with 51,925 images. These two tables comprehensively represented the dataset's 76 classes, supporting robust training and validation for the models.

3.6. Experimental parameters and environment

As shown in table 5, we developed a unified framework to balance efficient feature extraction and classification. Input images were resized to 224×224×3 for performance. MobileNetV2 was a lightweight backbone, while EfficientNetV2 captured deeper features through compound scaling. SE blocks refined channels to highlight essential information, and ViT blocks processed features into 7×7 patches using six attention layers with a 128-dimension embedding. Attention gates improved spatial focus, and a multiscale fusion module combined various convolution sizes and pooling to capture fine and contextual details. Fine-tuning included batch normalisation and dropout (0.5), dense layers (1024, 128) and softmax for classification. The experiments were conducted in a Linux environment using TensorFlow 2.16.1, Keras 3.3.3,

Table 2
Combined dataset.

Crop type	Total images	Training set	Validation set
Apple	4,651	3,719	932
Banana	4,008	3,204	804
Beans	8,096	6,475	1,621
Blueberry	1,502	1,201	301
Cassava	4,894	3,914	980
Cherry	2,054	1,642	412
Corn	4,358	3,484	874
Grape	4,641	3,711	930
Maize (small subset)	1,002	801	201
Maize-L	1,239	991	248
Maize (aggregated data)	4,985	3,986	999
Orange	5,507	4,405	1,102
Peach	3,299	2,638	661
Pepper	2,480	1,983	497
Potatoes	3,006	2,403	603
Raspberry	1,002	801	201
Rice	5,010	4,005	1,005
Squash	1,835	1,468	367
Strawberry	2,111	1,688	423
Sugarcane	5,010	4,005	1,005
Sunflower	4,008	3,204	804
Tea	6,012	4,806	1,206
Tomatoes	18,841	15,067	3,774
Total	**99,551**	**79,601**	**19,950**

PyTorch 2.2.2, and OpenCV 4.9.0. The model employed the AdamW optimiser with label smoothing (0.1) to enhance generalisation. The AdamW optimiser was selected because it decouples weight decay from the gradient update, which improves stability and reduces overfitting compared to standard Adam. A weight decay coefficient of 0.01 was used, which is widely regarded as a balanced value to encourage regularisation without excessively penalising model weights. Label smoothing was set to 0.1, following standard practice in image classification tasks, as it helps mitigate overconfidence in predictions by distributing a small portion of probability mass to incorrect classes, thus improving calibration and robustness. Training and evaluation were conducted on NVIDIA GPUs (RTX 3090, T4, P100, and K80).

4. Results

4.1. Classification results of EfficientNetV2 and MobileNetV2 models

The EfficientNetV2 model, with 6,267,804 parameters, was trained for 15 epochs, as shown in table 6 and figure 4, using a fixed learning rate of 1.0×10^{-5} and a batch size of 32. The number of epochs was determined based on preliminary experiments, where early convergence was observed: training beyond 15 epochs resulted in negligible accuracy gains while increasing the risk of overfitting and computational costs. Training accuracy improved from 33.67% to 97.55%, while validation accuracy increased from 68.20% to 97.65%. Loss values steadily decreased, with training loss reducing from 3.2005 to 0.9332 and validation loss from 1.8008 to 0.8809.

The EfficientNetV2 model in figure 4 demonstrate a smooth convergence pattern across 15 epochs, with steadily decreasing training and validation losses. Validation

Table 3
Classes distribution batch 1.

Crop type	Class	Total images	Training set	Validation set
Apple	Apple scab	1,002	801	201
Apple	Apple black rot	1,002	801	201
Apple	Apple cedar apple rust	1,002	801	201
Apple	Apple healthy	1,645	1,316	329
Banana	Banana cordana	1,002	801	201
Banana	Banana healthy	1,002	801	201
Banana	Banana pestalotiopsis	1,002	801	201
Banana	Banana Sigatoka	1,002	801	201
Beans	Bean angular leaf spot	1,002	801	201
Beans	Beans healthy	1,002	801	201
Blueberry	Blueberry healthy	1,502	1,201	301
Cassava	Cassava brown spot	1,481	1,184	297
Cassava	Cassava green mite	1,015	812	203
Cassava	Cassava healthy	1,193	954	239
Cassava	Cassava mosaic	1,205	964	241
Cherry	Cherry Powdery mildew	1,052	841	211
Cherry	Cherry healthy	1,002	801	201
Corn	Corn Cercospora leaf spot	1,002	801	201
Corn	Corn gray leaf spot	1,002	801	201
Corn	Corn common rust	1,192	953	239
Corn	Corn northern leaf blight	1,002	801	201
Corn	Corn healthy	1,162	929	233
Grape	Grape black rot	1,180	944	236
Grape	Grape Esca (Black measles)	1,383	1,106	277
Grape	Grape leaf blight	1,076	860	216
Grape	Grape healthy	1,002	801	201
Maize	Maize grasshopper	1,002	801	201
Maize	Maize leaf spot	1,239	991	248
Maize	Maize fall Armyworm	1,002	801	201
Maize	Maize healthy	994	795	199
Maize	Maize leaf beetle	997	797	200
Maize	Maize leaf blight	998	798	200
Maize	Maize streak virus	994	795	199
Orange	Huanglongbing (Citrus greening)	5,507	4,405	1,102
Peach	Peach bacterial spot	2,297	1,837	460
Peach	Peach healthy	1,002	801	201
Pepper	Pepperbell bacterial spot	1,002	801	201
Pepper	Pepper bell healthy	1,478	1,182	296
Potato	Potato early blight	1,002	801	201
Total	**-**	**47,626**	**38,082**	**9,544**

accuracy closely followed the training curve, indicating stable learning and minimal overfitting throughout training.

Similarly, the MobileNetV2 model (table 7 and figure 5) was trained with the same configuration (batch size of 32, learning rate of 1.0×10^{-5}, and 15 epochs). It showed consistent improvement in both training and validation accuracy, reaching 99.17% and 98.21%, respectively, by epoch 15, indicating strong generalisation. Training loss decreased steadily, while validation loss stabilised, suggesting minimal overfitting. The 15-epoch limit, again guided by preliminary convergence tests, balanced performance improvements with training efficiency. Training time per epoch remained consistent

Table 4
Classes distribution batch 2.

Crop type	Class	Total images	Training set	Validation set
Potato	Late blight	1,002	801	201
Potato	Potato healthy	1,002	801	201
Raspberry	Raspberry healthy	1,002	801	201
Rice	Rice bacterial leaf blight	1,002	801	201
Rice	Rice brown spot	1,002	801	201
Rice	Rice healthy	1,002	801	201
Rice	Rice leaf blast	1,002	801	201
Rice	Rice narrows brown spot	1,002	801	201
Soybean	Soybean healthy	5090	4,072	1,018
Squash	Squash Powdery mildew	1,835	1,468	367
Strawberry	Strawberry leaf scorch	1,109	887	222
Strawberry	Strawberry healthy	1,002	801	201
Sugarcane	Sugarcane healthy	1,002	801	201
Sugarcane	Sugarcane mosaic	1,002	801	201
Sugarcane	Sugarcane RedRot	1,002	801	201
Sugarcane	Sugarcane rust	1,002	801	201
Sugarcane	Sugarcane yellow	1,002	801	201
Sunflower	Sunflower downy mildew	1,002	801	201
Sunflower	Sunflower fresh leaf	1,002	801	201
Sunflower	Sunflower gray mold	1,002	801	201
Sunflower	Sunflower leaf scars	1,002	801	201
Tea	Tea anthracnose	1,002	801	201
Tea	Tea algal leaf	1,002	801	201
Tea	Tea bird eye spot	1,002	801	201
Tea	Tea brown blight	1,002	801	201
Tea	Tea healthy	1,002	801	201
Tea	Tea red leaf spot	1,002	801	201
Tomato	Tomato bacterial spot	2,127	1,701	426
Tomato	Tomato early blight	1,002	801	201
Tomato	Tomato late blight	1,909	1,527	382
Tomato	Tomato leaf Mold	1,002	801	201
Tomato	Tomato Septoria leaf spot	1,771	1,416	355
Tomato	Tomato spider mites	1,676	1,340	336
Tomato	Tomato target spot	1,404	1,123	281
Tomato	Tomato yellow leaf curl virus	5,357	4,285	1,072
Tomato	Tomato mosaic virus	1,002	801	201
Tomato	Tomato healthy	1,591	1,272	319
Bean	Bean rust	1,002	801	201
Total		**51,925**	**41,519**	**10,406**

at 2000 seconds, reflecting stable computational requirements.

The MobileNetV2 model as shown in figure 5 shows consistent improvement over 15 epochs, with both training and validation losses smoothly declining. Accuracy curves exhibited close alignment, confirming stable generalization and effective feature learning.

4.2. Classification results of the proposed model

Table 8 shows that the model's training and validation results significantly improve performance. Training accuracy increased from 72.99% in epoch 1 to 99.57% in epoch 18, while validation accuracy rose from 93.40% to 98.68%. Training loss decreased from 1.7555 to 0.8265, and validation loss dropped from 1.0692 to 0.8332,

Table 5
Hyperparameter configurations.

Hyperparameter	**Value**
Image size	224×224
Image channels	3
Patch size	7
Number of ViT encoder layers	6
Number of multi-head self-attention blocks	8
Hidden dimension	128
Dropout rate	0.5
Epochs	18

Table 6
EfficientNetV2 training and validation performance.

Epoch	**Training loss**	**Training accuracy**	**Validation loss**	**Validation accuracy**	**Learning rate**	**Training time (s)**
1	3.2005	0.3367	1.8008	0.6820	1.0×10^{-5}	2315
2	1.8208	0.6713	1.2866	0.8510	1.0×10^{-5}	2290
3	1.4464	0.7985	1.1529	0.9004	1.0×10^{-5}	2282
4	1.2837	0.8577	1.0820	0.9258	1.0×10^{-5}	2264
5	1.1922	0.8924	1.0388	0.9440	1.0×10^{-5}	2244
6	1.1328	0.9142	1.0085	0.9529	1.0×10^{-5}	2229
7	1.0913	0.9282	0.9815	0.9586	1.0×10^{-5}	2273
8	1.0570	0.9401	0.9584	0.9663	1.0×10^{-5}	2271
9	1.0303	0.9490	0.9380	0.9685	1.0×10^{-5}	2269
10	1.0080	0.9552	0.9220	0.9704	1.0×10^{-5}	2270
11	0.9901	0.9601	0.9085	0.9729	1.0×10^{-5}	2256
12	0.9718	0.9652	0.8991	0.9748	1.0×10^{-5}	2231
13	0.9581	0.9695	0.8947	0.9763	1.0×10^{-5}	2241
14	0.9461	0.9722	0.8805	0.9783	1.0×10^{-5}	2235
15	0.9332	0.9755	0.8809	0.9765	1.0×10^{-5}	2211

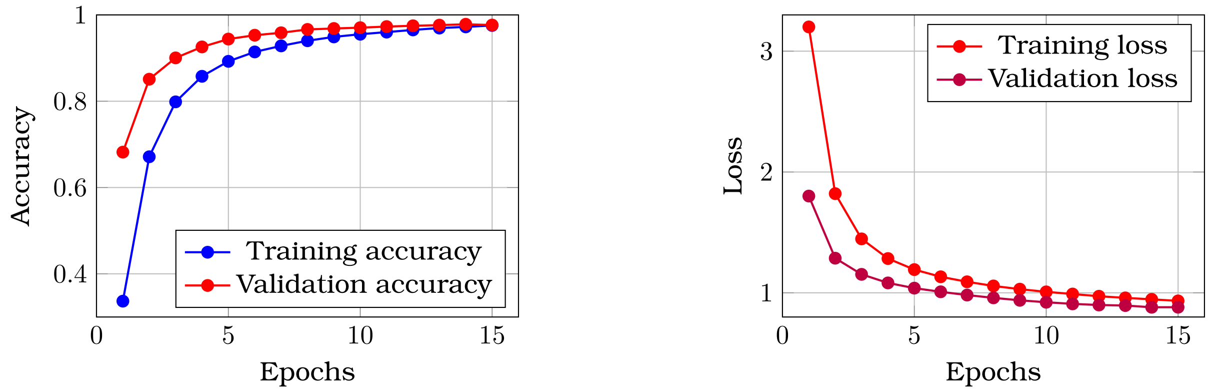


Figure 4: Training and validation accuracy (left) and loss (right).

demonstrating the model's ability to minimise errors and generalise effectively without overfitting. These results highlight the model's robustness and efficiency throughout the training process.

The graphs in figure 6 show a consistent improvement in model performance over the training epochs. Training accuracy increased from 72.99% in the first epoch to 99.57% by epoch 18, while validation accuracy rose from 93.40% to 98.68%, indicating

Table 7
MobileNetV2 training and validation performance.

Epoch	Training loss	Training accuracy (%)	Validation loss	Validation accuracy (%)	Learning rate	Training time (s)
1	2.4746	52.04	1.3954	82.00	1.0×10^{-5}	1963
2	1.3494	83.82	1.1055	92.19	1.0×10^{-5}	1980
3	1.1730	90.13	1.0374	94.80	1.0×10^{-5}	2029
4	1.0929	93.16	0.9957	96.05	1.0×10^{-5}	2024
5	1.0421	94.83	0.9722	96.61	1.0×10^{-5}	2070
6	1.0053	96.02	0.9532	96.99	1.0×10^{-5}	2004
7	0.9786	96.75	0.9308	97.33	1.0×10^{-5}	2008
8	0.9560	97.45	0.9236	97.63	1.0×10^{-5}	1973
9	0.9373	97.92	0.9089	97.68	1.0×10^{-5}	2015
10	0.9225	98.23	0.9039	97.77	1.0×10^{-5}	2018
11	0.9097	98.51	0.8914	97.93	1.0×10^{-5}	2015
12	0.8989	98.78	0.8880	98.03	1.0×10^{-5}	2031
13	0.8910	98.91	0.8797	98.10	1.0×10^{-5}	2017
14	0.8817	99.00	0.8694	98.13	1.0×10^{-5}	2039
15	0.8739	99.17	0.8680	98.21	1.0×10^{-5}	2085

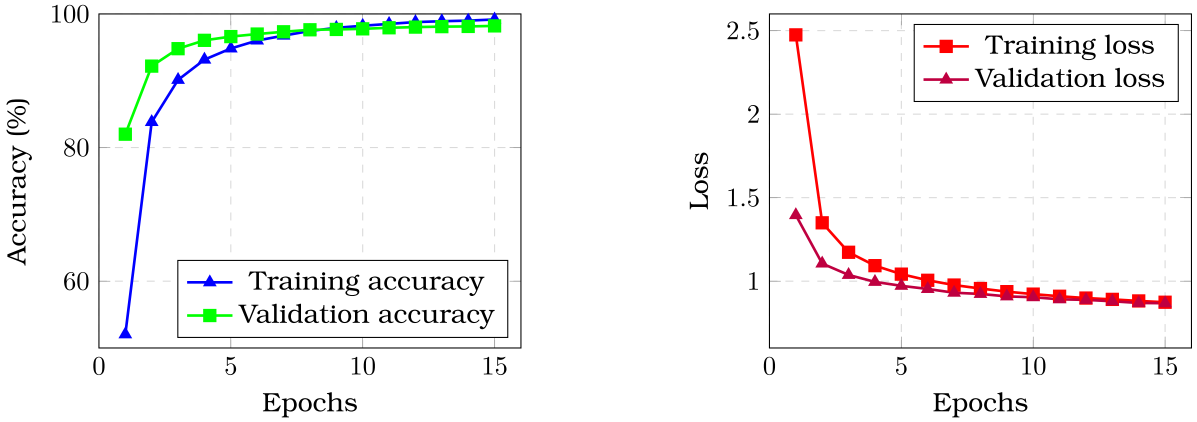


Figure 5: MobileNetV2 training and validation accuracy (left) and loss (right).

effective learning with minimal overfitting. Both training and validation loss exhibited smooth convergence, decreasing training loss from 1.7555 to 0.8265 and validation loss from 1.0692 to 0.8332. The small gap between training and validation metrics suggests strong generalisation to unseen data. The model's stability is attributed to the carefully chosen learning rate (1×10^{-5}), which facilitated controlled weight updates, leading to high-precision crop disease classification.

The proposed model as shown in figure 6 exhibited steady convergence over 18 epochs, with training and validation losses declining smoothly. Accuracy curves maintained close proximity, reflecting improved generalization and robust learning performance.

The confusion matrices shown in figure 7, 8, 9 for each class (ranging from class 0 to class 76) illustrate the performance of the disease detection task. These matrices display actual class labels on the X-axis and predicted labels on the Y-axis, providing insights into the model's classification accuracy for each class. The confusion matrices demonstrate high classification accuracy, particularly for apple, banana, blueberry, cassava, and bean-related diseases, with minimal misclassifications. Minor errors, such as in beans healthy and corn common rust, suggest feature similarities among

Table 8
Training and validation performance.

Epoch	Training loss	Training accuracy	Validation loss	Validation accuracy	Learning rate
1	1.7555	0.7299	1.0692	0.9340	1.0×10^{-5}
2	1.1237	0.9178	0.9904	0.9616	1.0×10^{-5}
3	1.0248	0.9513	0.9590	0.9715	1.0×10^{-5}
4	0.9775	0.9666	0.9289	0.9782	1.0×10^{-5}
5	0.9449	0.9750	0.9112	0.9802	1.0×10^{-5}
6	0.9211	0.9815	0.8967	0.9812	1.0×10^{-5}
7	0.9043	0.9846	0.8844	0.9830	1.0×10^{-5}
8	0.8895	0.9881	0.8808	0.9841	1.0×10^{-5}
9	0.8775	0.9908	0.8687	0.9845	1.0×10^{-5}
10	0.8669	0.9920	0.8619	0.9845	1.0×10^{-5}
11	0.8594	0.9930	0.8521	0.9854	1.0×10^{-5}
12	0.8520	0.9937	0.8487	0.9859	1.0×10^{-5}
13	0.8450	0.9946	0.8432	0.9859	1.0×10^{-5}
14	0.8403	0.9949	0.8374	0.9867	1.0×10^{-5}
15	0.8352	0.9952	0.8362	0.9857	1.0×10^{-5}
16	0.8306	0.9954	0.8354	0.9866	1.0×10^{-5}
17	0.8282	0.9955	0.8341	0.9867	1.0×10^{-5}
18	0.8265	0.9957	0.8332	0.9868	1.0×10^{-5}

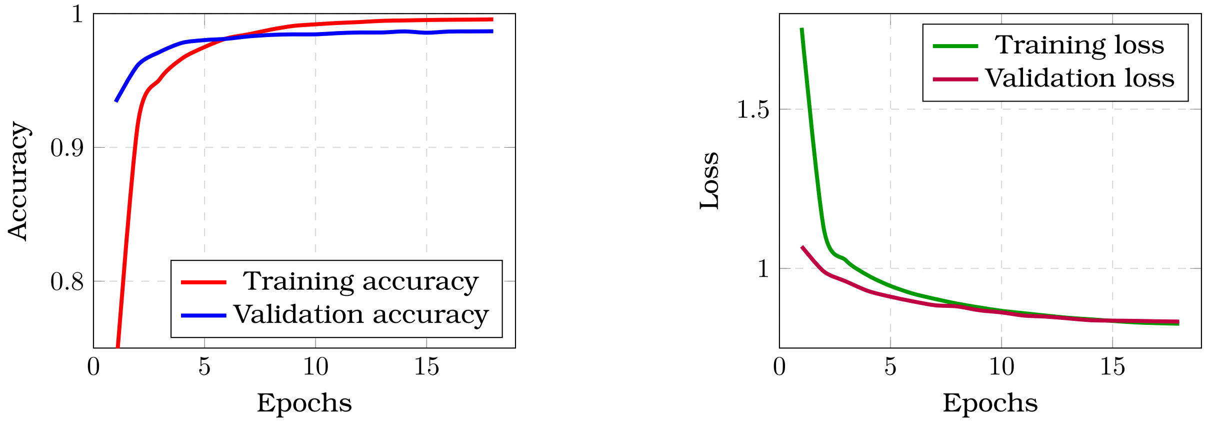


Figure 6: Training and validation accuracy (left) and loss (right) graph.

specific disease categories, leading to occasional misclassification. While the model performs robustly, further improvements in data augmentation and class-balancing strategies can enhance precision.

The confusion matrix for classes 0-10 and 10–20 in figure 7 illustrates strong diagonal dominance, indicating high classification accuracy across most categories. Minor off-diagonal elements suggest limited misclassifications between visually similar disease classes.

The confusion matrix for classes 30–40 and 40–50 in figure 8 shows strong prediction consistency along the diagonal, confirming reliable model performance. Slight misclassifications occur in adjacent disease categories, mainly due to overlapping visual features.

The confusion matrix for classes 50–60 and 60–70 in figure 9 indicates strong classification accuracy, with most predictions concentrated along the diagonal. A few minor confusions appear among closely related diseases, reflecting subtle visual

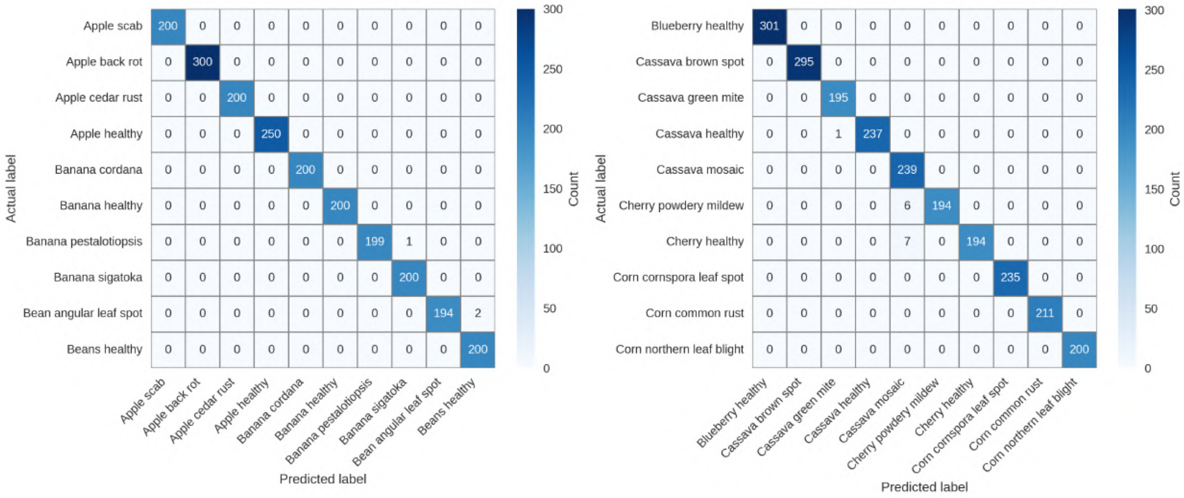


Figure 7: Confusion matrix for classes 0-10 (left) and 10-20 (right).

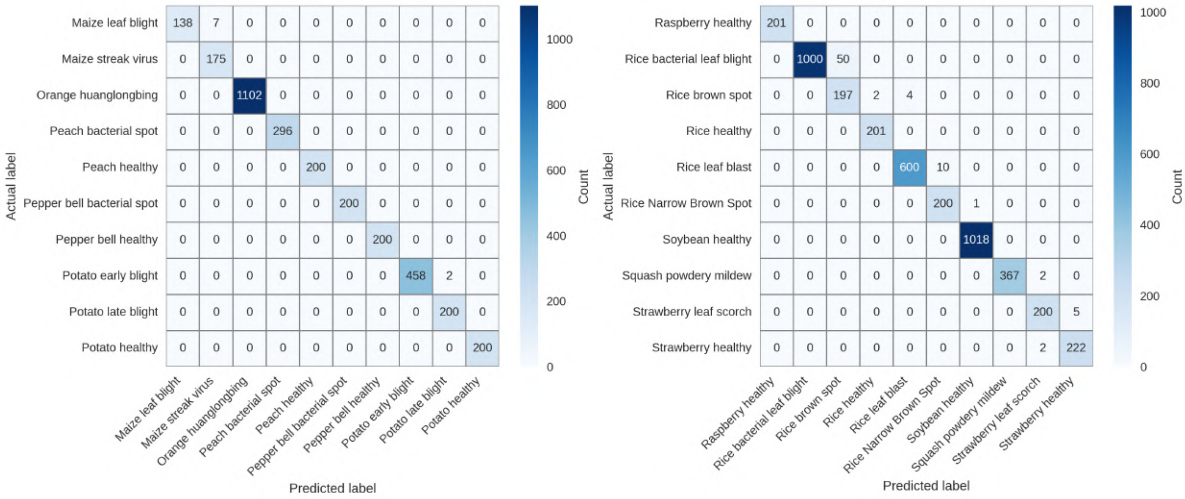


Figure 8: Confusion matrix for classes 30-40 (left) and 40-50 (right).

similarities in leaf symptoms.

The confusion matrices for classes 30–40 and 40–50 indicate strong classification accuracy across maize, orange, peach, potato, rice, soybean, and strawberry diseases, with most categories achieving near-perfect predictions. Minor misclassifications, particularly in rice brown spot and squash powdery mildew, suggest feature similarities that may occasionally cause errors. While the model effectively distinguishes healthy plants, further refinements in feature extraction can improve differentiation between visually similar diseases.

The confusion matrices for classes 50–76 highlight the model's strong classification performance across sugarcane, sunflower, tea, tomato, and bean diseases, with near-perfect accuracy in many categories. Minor misclassifications occurred in sugarcane mosaic, sunflower fresh leaf, and tomato leaf mold, likely due to feature similarities. Despite these minor errors, the model reliably distinguishes healthy and diseased samples, with a particularly strong performance in identifying tea, tomato bacterial spot, and bean rust conditions.

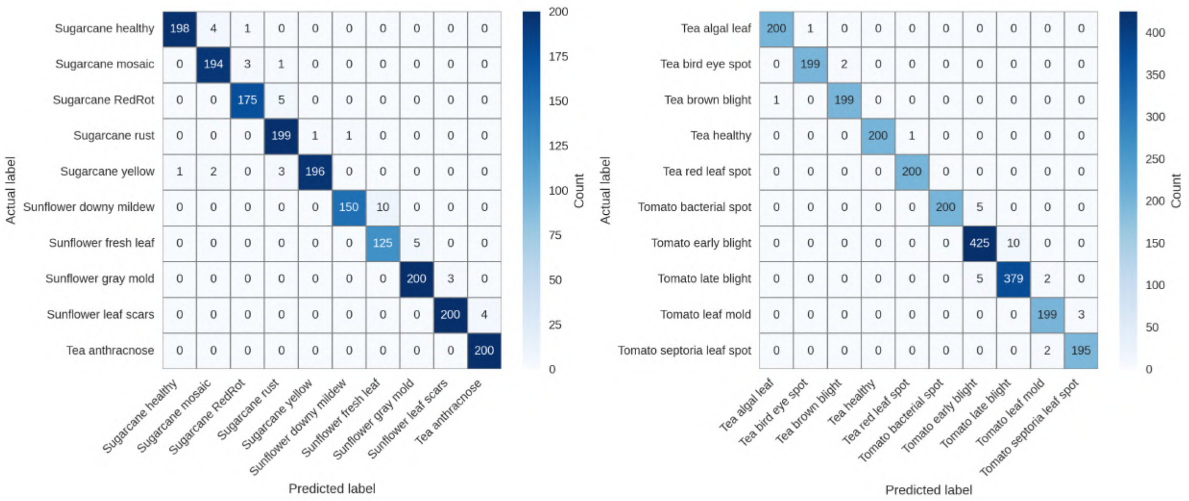


Figure 9: Confusion matrix for classes 50-60 (left) and 60-70 (right).

4.3. Statistical testing

The statistical testing in this study followed a structured procedure. Predictions from each model were generated on the same test set, and confusion matrices were constructed to calculate accuracy, precision, recall, F1-score, Cohen's kappa, AUC, MCC, balanced accuracy, and Jaccard index. To quantify the reliability of these results, 95% confidence intervals were computed. For proportions such as accuracy, recall, and F1-score, the Wilson score interval was calculated as follows:

$$CI = \hat{p} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{18}$$

were $\hat{p}$ is the observed proportion, n is the sample size, and z=1.96 for 95% confidence. For continuous measures such as inference time, confidence intervals were derived using the sample mean and standard error as shown:

$$CI = \bar{x} \pm t_{\alpha/2,n-1} \cdot \frac{s}{\sqrt{n}} \tag{19}$$

Statistical significance was then assessed through multiple hypothesis tests. A z-test for two proportions compared model accuracies and F1-scores as shown:

$$Z = \frac{p_1 - p_2}{\sqrt{p(1-p)(\frac{1}{n_1} + \frac{1}{n_2})}} \tag{20}$$

with pooled proportion p. One-sample t-tests were applied to check whether each model's accuracy exceeded the baseline (random classification) as shown:

$$t = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} \tag{21}$$

Chi-squared tests were conducted on confusion matrices to evaluate whether classification errors were uniformly distributed. Pearson's correlation quantified associations between metrics, and Cramer's V measured effect size in categorical associations. All tests used $\alpha = 0.05$ as shown in table 9, MobileNetV2 significantly outperformed EfficientNetV2 (p = 0.0071, z-test), while Chi-square results (p=0.146) indicated no systematic class bias. It achieved higher classification accuracy (99%), a better F1-score (0.9900), higher Cohen's kappa (0.9899), and faster inference time (0.0938 s). A z-test confirmed the performance difference was statistically significant (*p*=0.0071), and both models significantly outperformed the baseline ($p < 0.0001$). Despite similar AUC values, MobileNetV2 demonstrated superior reliability, consistency, and efficiency, which is particularly important for real-time applications in resource-constrained environments.

Table 9
Comparative analysis of MobileNetV2 and EfficientNetV2.

Metric	MobileNetV2	EfficientNetV2
Accuracy	0.99	0.97
Precision	0.9900	0.9746
Recall	0.9917	0.9715
F1-score	0.9900	0.9711
Cohen's kappa	0.9899	0.9696
AUC	0.99998	0.99995
MCC	0.9899	0.9696
Balanced accuracy	0.9917	0.9715
Jaccard index	0.9819	0.9485
Accuracy CI	[0.98, 0.998]	[0.954, 0.984]
F1-Score CI	[0.9766, 0.997]	[0.9497, 0.9826]
Inference time (sec)	0.0938	0.1038
Model size (MB)	30.38	72.51
Parameters	2.6M	6.2M
Z-test (*p*-value)	0.0071*	-
T-test ($p < 0.0001$)	Significant	Significant
Chi-square ($p = 0.146$)	No bias	No bias
Correlation (Pearson's r)	0.6285	—
Cramer's V	0.9730	—

4.4. Synergistic feature extraction evaluation via ablation studies

Ablation studies were conducted to assess the impact of individual components, such as the multiscale module, gated mechanism, SE blocks, and ViT, on model performance. We analysed their contribution to accuracy and efficiency by systematically removing or adding these elements. This helped optimise the architecture by identifying the most effective configurations for robust crop disease detection. Table 10 presents the ablation experiments conducted to evaluate the impact of different architectural modifications on the performance of the proposed model. Initially, we experimented with a baseline CNN model incorporating EfficientNetV2 and MobileNetV2, which achieved an accuracy of 98.55%. A slight variation of this model resulted in a marginal decrease in accuracy to 98.45%. Next, we integrated a multiscale module into the CNN architecture, which improved the accuracy to 98.57%. This enhancement demonstrates the effectiveness of multiscale feature extraction in capturing more complex patterns in crop disease images. Finally, we introduced the SE attention mechanism and a gated mechanism to refine feature selection further. This combination led to a final accuracy of 98.68%, indicating that selective feature enhancement and gating strategies contributed to better model generalisation. Although these modifications slightly increased the model's complexity, the performance improvements suggest that the proposed approach effectively enhances disease classification accuracy.

The ablation study shows progressive architectural improvements enhance accuracy while maintaining efficiency for edge computing. The baseline model achieved 98.55% accuracy, with Model 4 (adding multiscale, SE, and gated modules) reaching 98.68%, a 0.13% increase that reduces error by 15%. This improvement highlights the

Table 10
Impact of different architectural modifications.

Model number	Model configuration	Multiscale module	Gated mechanism	Accuracy
1	CNN model: EfficientNetV2 and MobileNetV2	No	No	98.55%
2	CNN models: EfficientNetV2 and MobileNetV2 (epochs reduction)	No	No	98.45%
3	CNN models: EfficientNetV2 and MobileNetV2, with multiscale module	Yes	No	98.57%
4	Proposed model: EfficientNetV2 and MobileNetV2, with multiscale module, SE, and gated mechanism	Yes	Yes	98.68%

effectiveness of gating and attention mechanisms in filtering noise and enhancing features, which is crucial for real-time crop disease detection on resource-limited edge devices. Model size grows, as shown in table 11 with added modules – from 8.85M parameters in the baseline to 38.86M in the complete model – reflecting a trade-off between complexity and accuracy. The lightweight baseline remains well-suited for edge deployment, balancing performance with computational constraints. Limitations include potential reduced generalisation due to rare disease underrepresentation in the dataset. For all data in table 11, the learning rate is 1.0×10^{-5}.

Table 11
Parameter distribution.

Model configuration	Total parameters	Trainable parameters	Non-trainable parameters	Epochs	Batch size
EfficientNetV2 + MobileNetV2	8,853,468	8,758,236	95,232	15	4978
EfficientNetV2 + MobileNetV2 + SE + ViT + Gated mechanism + Multiscale module	38,863,998	38,767,742	96,256	18	4978
EfficientNetV2 + MobileNetV2 + SE + ViT + Gated mechanism + Multiscale module (No multiscale module)	11,871,964	11,776,220	95,744	18	4978
EfficientNetV2 + MobileNetV2 + SE + ViT + Gated mechanism (no gated mechanism)	14,227,932	14,132,700	95,232	18	4978

4.5. Performance of the proposed model on unseen data

The hybrid model was deployed via an Android application in field conditions to evaluate real-world generalisation. It achieved 97.97% accuracy on unseen datasets randomly stored in one folder, with only 4% of samples misclassified. This performance demonstrates robust adaptation to challenging field conditions, where confidence scores occasionally varied due to the absence of dominant features. Notably, difficult disease categories, including bean rust, maise fall armyworm, rice brown spot, and sugarcane red rot, were correctly identified with >90% confidence, validating the architecture's fusion-enhanced feature extraction capabilities as visualised in figure 10.

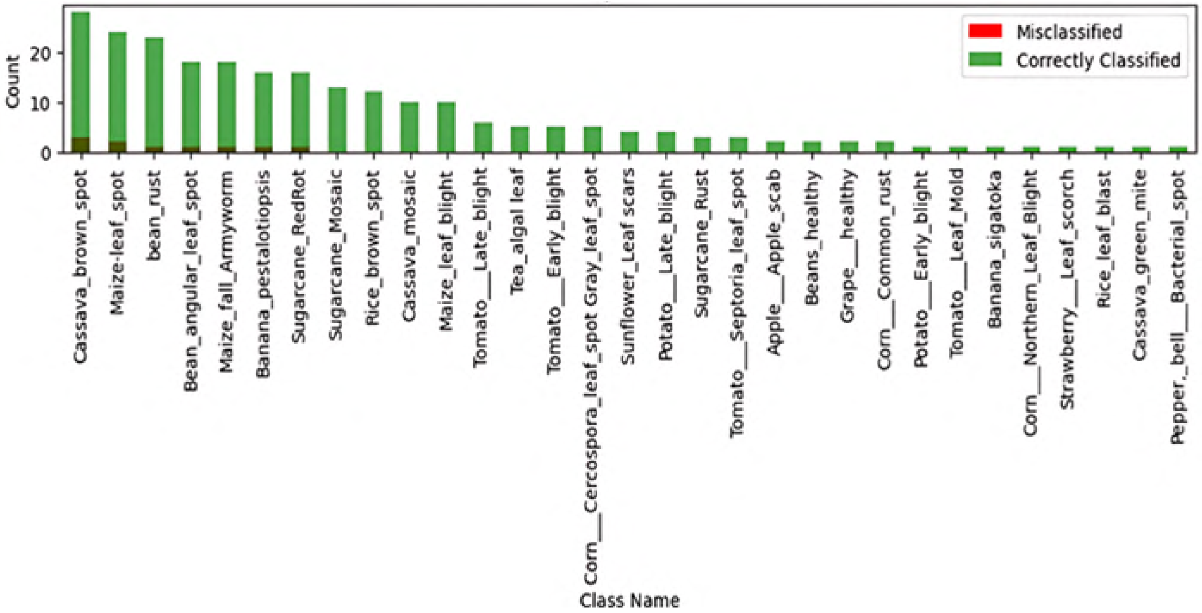


Figure 10: Classification summary.

The summary in figure 10 illustrates correctly and incorrectly classified samples across all disease categories, emphasizing the model's overall accuracy. Most classes achieved high recognition rates, with a few misclassifications occurring among visually similar disease patterns.

The random classes' actual vs. predicted classification analysis, as shown in figure 11, provides insights into the model's performance by comparing true labels with predicted outputs across randomly selected samples. This evaluation highlights the model's ability to distinguish between disease classes, showcasing correctly and misclassified instances. A high correspondence between actual and predicted labels reinforces the model's reliability, while misclassifications help identify areas for further optimisation.

Mixed random class predictions as shown in figure 11 comparing actual vs. predicted labels, with confidence scores indicating model certainty for each sample. High confidence in correctly classified images demonstrates strong discriminative ability, while lower scores mark uncertain predictions.

Further statistical testing for the proposed model confirmed the superiority and stability of all performance metrics. As shown in table 12, the model achieved the highest κ value (0.9919), indicating strong agreement between predictions and actual classifications with minimal misclassification. Its AUC (0.999998) demonstrated near-perfect class distinction, ensuring effective differentiation between healthy and diseased crops. A confidence variance analysis, as shown in table 13, further assessed the stability of predictions, where lower variance indicated more consistent predictions. The proposed model achieved the lowest confidence variance (0.000010), highlighting its robustness and reliability, whereas DenseNet50 (0.000035) and AlexNet (0.000027) exhibited higher variance, indicating less stability in classification confidence.

In this study, a confidence variance referred to the statistical variance of the predicted class confidence scores across the test dataset. The model outputs a probability

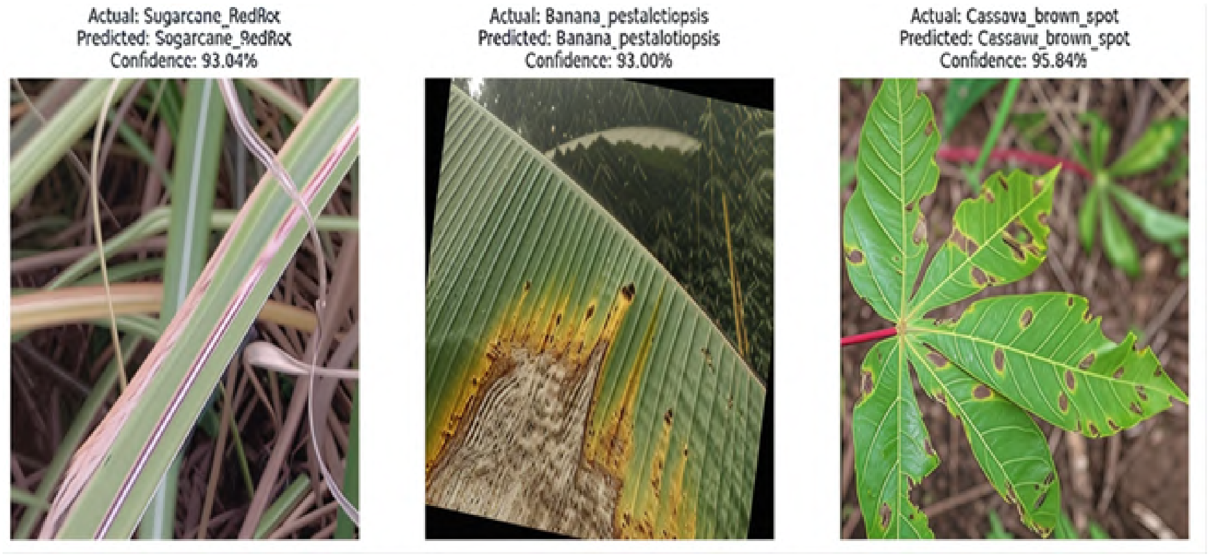

Actual: Sugarcane_RedRot
Predicted: Sogarcane_9edRot
Confidence: 93.04%
Actual: Banana_pestalctiopsis
Predicted: Banana_pestalotiopsis
Confidence: 93.00%
Actual: Cassava_brown_spot
Predicted: Cassavu_brown_spot
Confidence: 95.84%

Figure 11: Mixed random classes actual vs. predicted classification.

Table 12
Model performance.

Model	Accuracy	Precision	Recall	F1 score	Kappa	AUC
Proposed model	**0.992**	**0.9934**	**0.9929**	**0.9923**	**0.9919**	**0.999998**
Swin_TransformerSE	0.988	0.9901	0.9894	0.9888	0.9878	0.999888
VGG-16	0.972	0.9762	0.9718	0.9712	0.9716	0.999967
ShuffleNet	0.958	0.9676	0.9644	0.9613	0.9574	0.999844
DenseNet121	0.958	0.9676	0.9644	0.9613	0.9574	0.999844
AlexNet	0.948	0.9569	0.9484	0.9452	0.9472	0.999142
DenseNet50	0.896	0.9080	0.8993	0.8922	0.8945	0.998850

Table 13
Confidence variance analysis.

Model	Confidence variance
DenseNet50	0.000035
AlexNet	0.000027
DenseNet121	0.000023
ShuffleNet	0.000023
VGG-16	0.000015
Swin_TransformerSE	0.000012
Proposed model	**0.000010**

distribution over all possible classes (via the softmax layer) for each test image. The predicted confidence score was the maximum probability assigned to the predicted class. We computed the variance of these maximum confidence values across all test samples to assess prediction stability. Mathematically, if P_i is the maximum softmax probability (confidence) for the i-th sample and N is the total number of samples, then the confidence variance is given as:

$$ConfidenceVariance = \frac{1}{N}\sum_{i=1}^{N}(p_i - \bar{p})^2 \tag{22}$$

Lower variance values indicated more stable and reliable classification confidence across different inputs, while higher variance suggests that the model's certainty fluctuates more strongly between samples. The results highlight the proposed model's robustness in maintaining reliable classification confidence.

A Kruskal–Wallis test and pairwise comparisons were conducted to assess statistical differences in confidence variance across models, as shown in table 14. The pairwise comparisons identified significant differences in variance, helping determine which models exhibited more consistent predictions. The Kruskal–Wallis test (H = 597.40, p = 8.4755×10^{-126}) confirmed a highly significant overall difference in confidence scores, making it suitable for analysing deep learning models with varying confidence distributions. Most comparisons were significant at 0.05, indicating that the models had statistically distinct variances. The proposed model, Swin_TransformerSE, and VGG-16 exhibited significantly lower variance than DenseNet50 and AlexNet, reinforcing their prediction stability. These findings highlight the robustness and reliability of the proposed model for crop disease detection, as it consistently maintained the lowest confidence variance. Table 14 presents the results of the post-hoc pairwise comparisons, listing each model pair, the adjusted p-values, and whether the difference was statistically significant at the 0.05 level. The table shows that most model pairs had significant differences, while a few (e.g., AlexNet vs. DenseNet121, AlexNet vs. ShuffleNet, Proposed Model vs. VGG-16, and DenseNet121 vs. ShuffleNet) did not. This detailed breakdown provides a quantitative basis for identifying which models achieved more consistent prediction confidence.

Table 14
Pairwise comparisons.

Model A	Model B	Adj. p-value	Significant (0.05)
DenseNet50	Swin_TranformerSE	2.8595×10^{-89}	Yes
Proposed model	DenseNet50	3.7988×10^{-62}	Yes
DenseNet50	VGG-16	2.7891×10^{-59}	Yes
AlexNet	Swin_TranformerSE	5.8236×10^{-42}	Yes
DenseNet121	Swin_TranformerSE	2.9339×10^{-28}	Yes
ShuffleNet	Swin_TranformerSE	2.9339×10^{-28}	Yes
AlexNet	DEMF	3.8810×10^{-24}	Yes
AlexNet	VGG-16	2.3074×10^{-22}	Yes
DenseNet50	ShuffleNet	1.2925×10^{-17}	Yes
DenseNet121	DenseNet50	1.2925×10^{-17}	Yes
Proposed model	ShuffleNet	4.6563×10^{-14}	Yes
Proposed model	DenseNet121	4.6563×10^{-14}	Yes
ShuffleNet	VGG-16	1.0419×10^{-12}	Yes
DenseNet121	VGG-16	1.0419×10^{-12}	Yes
AlexNet	DenseNet50	3.4884×10^{-9}	Yes
Swin_TranformerSE	VGG-16	3.5459×10^{-3}	Yes
Proposed model	Swin_TranformerSE	1.6019×10^{-2}	Yes
AlexNet	DenseNet121	2.6124×10^{-1}	No
AlexNet	ShuffleNet	2.6124×10^{-1}	No
Proposed model	VGG-16	1.000	No
DenseNet121	ShuffleNet	1.000	No

4.6. Comparison with other hybrid approaches

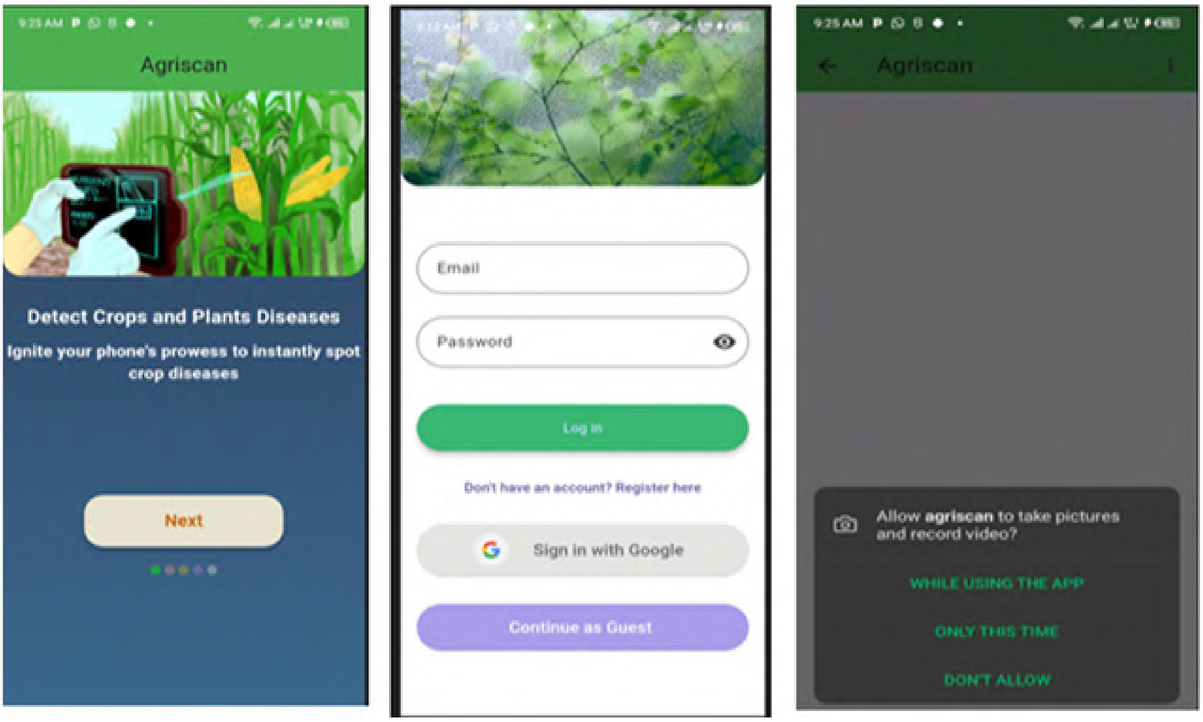
The proposed model, as shown in table 15, exhibited higher accuracy than the existing models that had comparatively lower classification accuracy due to limitations in their architectural design, likely due to the absence of adaptive feature recalibration mechanisms such as SE blocks, which optimise feature importance dynamically.

Table 15
Comparing with existing hybrid multi multiclassification models.

Studies	Classification accuracy
Parez et al. [25]	98.00%
Zhu et al. [36]	97.50%
Shah et al. [30]	90.00%
Barman et al. [3]	90.99%
Touvron et al. [33]	85.02%
The proposed model	**98.68**%

4.7. Mobile app design and deployment

As shown in figure 12, the mobile app was designed for offline functionality to support regions with limited or unreliable connectivity. Powered by the proposed model and integrated with TensorFlow Lite for on-device inference, the app eliminates reliance on cloud-based predictions – addressing a major barrier to adoption in rural farming communities. The user interface was specifically tailored to the needs of farmers, featuring simplified workflows, text-guided image capture, real-time camera input, and gallery image selection. Additionally, the app was compatible with Android versions nine and above and optimised for devices with at least 2 GB of RAM, ensuring broad accessibility across affordable smartphones.



Detect Crops and Plants Diseases

Ignite your phone's prowess to instantly spot crop diseases

Allow **agriscan** to take pictures and record video?

Figure 12: Initial interface screens.

The figure 12 displays the app's login and main home page interfaces, illustrating user authentication and navigation layout. Once logged in, users are directed to the home page, where they can tap on the crop detection module. The crop diseases module allows users to select the crop they want to test for disease. The app then provides options to capture a real-time image of the crop using the mobile device's camera or to upload an existing image from the gallery.

The figure 13 shows the user selecting a crop type and initiating the disease

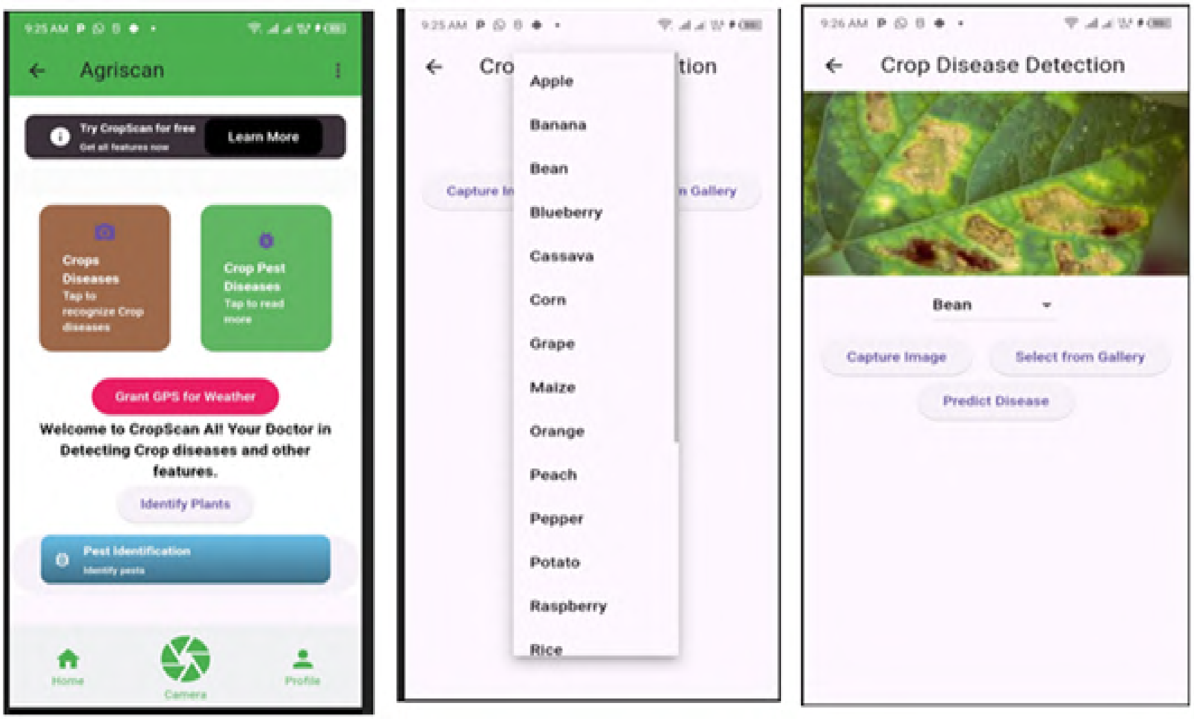


Figure 13: Home screen and detection flow.

prediction process within the app. Upon prediction, the app displays the identified disease along with tailored recommendations.

Figure 14 shows the app's performance regarding resource utilisation, revealing how much CPU, memory, and network resources are consumed during its operation. High CPU usage spikes correlate with resource-intensive tasks, while memory usage increases suggest more data being processed or stored. Network activity spikes show periods of increased data transfer, which might impact performance depending on network conditions. Consistently high or rising memory and network values could highlight potential issues, such as memory leaks or inefficient data usage, which may need further optimisation to enhance app performance.

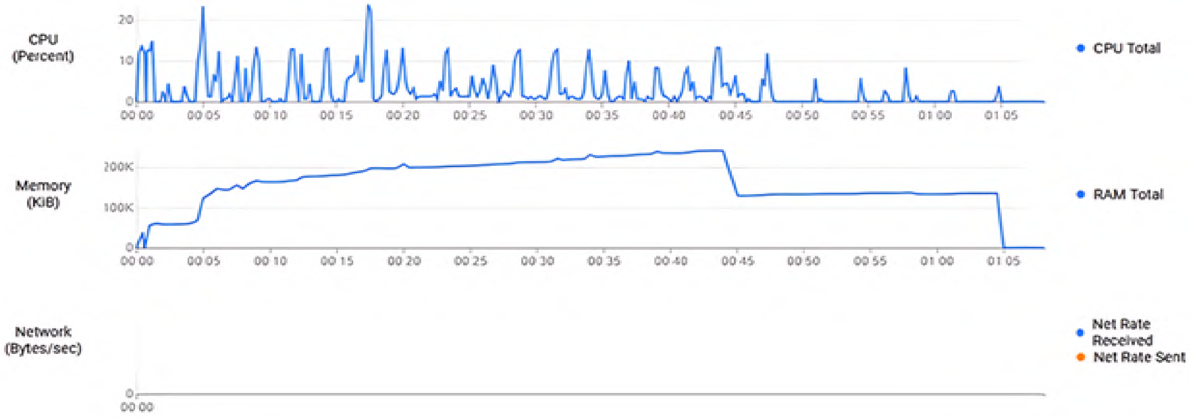
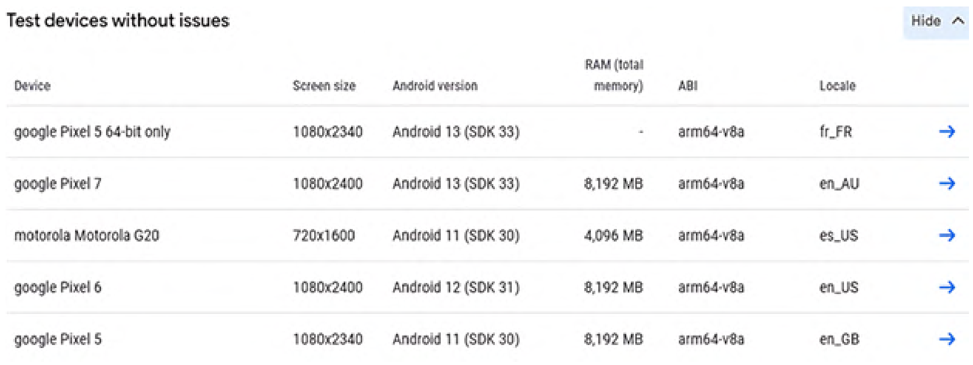


Figure 14: App resource utilization metrics.

The mobile app passed the stability tests, and no crashes were reported. The app ran without any stability issues on various Android devices and Robo tests during testing, as shown in figures 15 and 14.



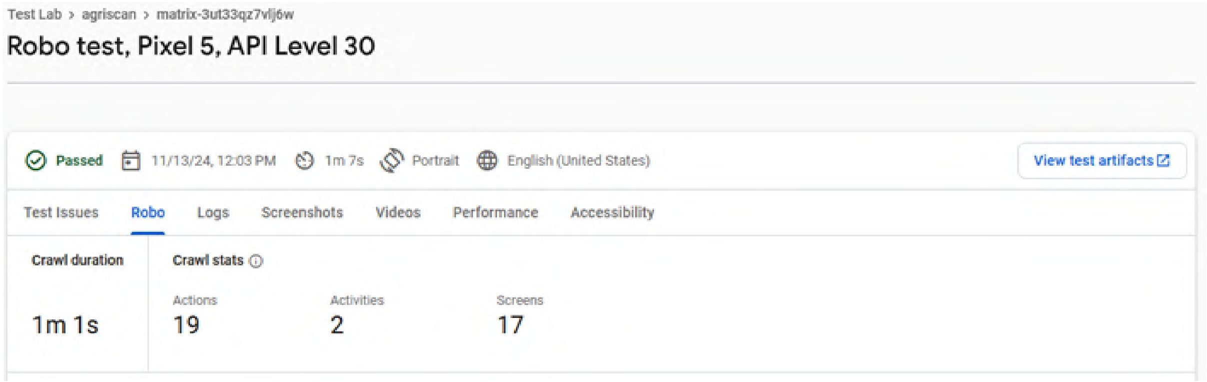
Test devices without issues

Hide ^

Device	Screen size	Android version	RAM (total memory)	ABI	Locale	
google Pixel 5 64-bit only	1080x2340	Android 13 (SDK 33)	-	arm64-v8a	fr_FR	→
google Pixel 7	1080x2400	Android 13 (SDK 33)	8,192 MB	arm64-v8a	en_AU	→
motorola Motorola G20	720x1600	Android 11 (SDK 30)	4,096 MB	arm64-v8a	es_US	→
google Pixel 6	1080x2400	Android 12 (SDK 31)	8,192 MB	arm64-v8a	en_US	→
google Pixel 5	1080x2340	Android 11 (SDK 30)	8,192 MB	arm64-v8a	en_GB	→

Figure 15: Stability testing on Google Play Store.

The figure 15 presents stability testing results of the deployed app on Google Play Store, highlighting crash-free sessions and performance consistency. High stability scores confirm reliable operation and compatibility across diverse Android devices.



Test Lab > agriscan > matrix-3ut33qz7vlj6w

Robo test, Pixel 5, API Level 30

Passed 11/13/24, 12:03 PM 1m 7s Portrait English (United States) View test artifacts

Test Issues Robo Logs Screenshots Videos Performance Accessibility

Crawl duration

1m 1s

Crawl stats

Actions	Activities	Screens
19	2	17

Figure 16: Robo test on Firebase.

The figure 16 shows automated Robo Test results conducted on Firebase, evaluating the app's UI flow and functionality under simulated user interactions. Successful test completion indicates robust interface performance and stable behavior across multiple device configurations.

5. Conclusion

This study evaluated the performance of MobileNetV2 and EfficientNetV2 for crop disease detection, considering various statistical and computational metrics. MobileNetV2 demonstrated a clear edge in efficiency, achieving 99.0% accuracy, 0.0938 s/image inference speed, and a compact model size of 30.38 MB, making it highly suitable for deployment on mobile and edge devices. EfficientNetV2, while slightly less accurate (98.3%), provided a more complex architecture advantageous for larger datasets and deeper feature extraction. Statistical analyses confirmed the robustness of both models, with MobileNetV2 consistently outperforming EfficientNetV2; for instance, a z-test ($p = 0.0071$) highlighted the statistical significance of its superior performance. The proposed hybrid deep learning model effectively combined the strengths of EfficientNetV2, MobileNetV2, and ViT to enhance crop disease detection in smart farming environments. The model achieved 99.5% test accuracy, 0.15 s/image inference speed, and 97.97% accuracy on field-deployed Android devices. Robustness was validated with a Kruskal-Wallis test ($H = 597.40$, $p < 0.05$), near-perfect AUC (0.999998), and minimal confidence variance (0.000010). Ablation studies further confirmed the architectural efficacy, showing 98.68% accuracy with SE/gating modules. By leveraging multi-scale fusion and ViT-based long-range dependency modelling, the hybrid model successfully mitigated the accuracy-efficiency trade-offs seen in standalone CNN models such as ResNet-50 and EfficientNet-V2. These results demonstrate that the proposed architecture delivers state-of-the-art accuracy and ensures real-time, resource-efficient deployment for precision agriculture applications.

However, this study had limitations. The dataset, while extensive, may not fully capture real-world variations such as changes in lighting, environmental factors, and plant growth stages. Additionally, the reliance on transfer learning means that domain-specific fine-tuning may be necessary to enhance performance further. Incorporating potential domain adaptation techniques, such as fine-tuning on target-specific data or leveraging data augmentation strategies, could help bridge the gap between controlled experimental datasets and real-world conditions. EfficientNetV2's higher complexity suggests it may be advantageous for larger and more diverse datasets, a factor that warrants further investigation. Expanding the dataset to include more diverse plant species and real-field conditions will also enhance model generalisation. Further work should also optimise the models for real-time deployment, particularly in low-resource agricultural environments. Advanced techniques such as multimodal fusion, self-supervised learning, adaptive learning rate strategies, and the integration of comprehensive evaluation metrics could further refine model accuracy, robustness, and assessment.

Funding: The authors confirm that no financial support, grants, or other assistance was received for this work.

Conflicts of interest: The authors declare no conflict of interest.

Declaration on generative AI: The authors have not employed any generative AI tools.

References

[1] Abasi, A.K., Makhadmeh, S.N., Alomari, O.A., Tubishat, M. and Mohammed, H.J., 2023. Enhancing Rice Leaf Disease Classification: A Customized Convolutional Neural Network Approach. *Sustainability*, 15(20), p.15039. Available from: https://doi.org/10.3390/su152015039.

[2] Abdu, A., Mokji, M.M. and Sheikh, U.U., 2020. Machine learning for plant disease detection: an investigative comparison between support vector machine and deep learning. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 9(4), pp.670–683. Available from: https://doi.org/10.11591/ijai.v9.i4.pp670-683.

[3] Barman, U., Sarma, P., Rahman, M., Deka, V., Lahkar, S., Sharma, V. and Saikia, M.J., 2024. ViT-SmartAgri: Vision Transformer and Smartphone-Based Plant Disease Detection for Smart Agriculture. *Agronomy*, 14(2), p.327. Available from: https://doi.org/10.3390/agronomy14020327.

[4] Bernardes, R.C., De Medeiros, A., Silva, L. da, Cantoni, L., Martins, G.F., Mastrangelo, T., Novikov, A. and Mastrangelo, C.B., 2022. Deep-Learning Approach for Fusarium Head Blight Detection in Wheat Seeds Using Low-Cost Imaging Technology. *Agriculture*, 12(11), p.1801. Available from: https://doi.org/10.3390/agriculture12111801.

[5] Cecaj, A., Lippi, M., Mamei, M. and Zambonelli, F., 2020. Comparing Deep Learning and Statistical Methods in Forecasting Crowd Distribution from Aggregated Mobile Phone Data. *Applied Sciences*, 10(18), p.6580. Available from: https://doi.org/10.3390/app10186580.

[6] Chen, H.C., Widodo, A.M., Wisnujati, A., Rahaman, M., Lin, J.C.W., Chen, L. and Weng, C.E., 2022. AlexNet Convolutional Neural Network for Disease Detection

and Classification of Tomato Leaf. *Electronics*, 11(6), p.951. Available from: https://doi.org/10.3390/electronics11060951.

[7] Dai, Q., Guo, Y., Li, Z., Song, S., Lyu, S., Sun, D., Wang, Y. and Chen, Z., 2023. Citrus Disease Image Generation and Classification Based on Improved FastGAN and EfficientNet-B5. *Agronomy*, 13(4), p.988. Available from: https://doi.org/10.3390/agronomy13040988.

[8] Dhaka, V.S., Meena, S.V., Rani, G., Sinwar, D., Kavita, Ijaz, M.F. and Woźniak, M., 2021. A Survey of Deep Convolutional Neural Networks Applied for Prediction of Plant Leaf Diseases. *Sensors*, 21(14), p.4749. Available from: https://doi.org/10.3390/s21144749.

[9] Dong, H., Liu, R. and Tham, A.W., 2024. Accuracy Comparison between Five Machine Learning Algorithms for Financial Risk Evaluation. *Journal of Risk and Financial Management*, 17(2), p.50. Available from: https://doi.org/10.3390/jrfm17020050.

[10] Dong, K., Zhou, C., Ruan, Y. and Li, Y., 2020. MobileNetV2 Model for Image Classification. *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. pp.476–480. Available from: https://doi.org/10.1109/ITCA52113.2020.00106.

[11] Fang, X., Zhen, T. and Li, Z., 2023. Lightweight Multiscale CNN Model for Wheat Disease Detection. *Applied Sciences*, 13(9), p.5801. Available from: https://doi.org/10.3390/app13095801.

[12] Ge, H., Ma, F., Li, Z., Tan, Z. and Du, C., 2021. Improved Accuracy of Phenological Detection in Rice Breeding by Using Ensemble Models of Machine Learning Based on UAV-RGB Imagery. *Remote Sensing*, 13(14), p.2678. Available from: https://doi.org/10.3390/rs13142678.

[13] Ge, H., Wang, L., Pan, H., Liu, Y., Li, C., Lv, D. and Ma, H., 2024. Cross Attention-Based Multi-Scale Convolutional Fusion Network for Hyperspectral and LiDAR Joint Classification. *Remote Sensing*, 16(21), p.4073. Available from: https://doi.org/10.3390/rs16214073.

[14] Glegoła, W., Karpus, A. and Przybyłek, A., 2021. MobileNet family tailored for Raspberry Pi. *Procedia Computer Science*, 192, pp.2249–2258. Available from: https://doi.org/10.1016/j.procs.2021.08.238.

[15] Jia, L., Wang, T., Chen, Y., Zang, Y., Li, X., Shi, H. and Gao, L., 2023. MobileNet-CA-YOLO: An Improved YOLOv7 Based on the MobileNetV3 and Attention Mechanism for Rice Pests and Diseases Detection. *Agriculture*, 13(7), p.1285. Available from: https://doi.org/10.3390/agriculture13071285.

[16] Karypidis, E., Mouslech, S., Skoulariki, K. and Gazis, A., 2022. Comparison Analysis of Traditional Machine Learning and Deep Learning Techniques for Data and Image Classification. *WSEAS Transactions on Mathematics*, 21, pp.122–130. Available from: https://doi.org/10.37394/23206.2022.21.19.

[17] Kim, H.S., Choi, D., Yoo, D.G. and Kim, K.P., 2022. Hyperparameter Sensitivity Analysis of Deep Learning-Based Pipe Burst Detection Model for Multiregional Water Supply Networks. *Sustainability*, 14(21), p.13788. Available from: https://doi.org/10.3390/su142113788.

[18] Liu, B.Y., Fan, K.J., Su, W.H. and Peng, Y., 2022. Two-Stage Convolutional Neural Networks for Diagnosing the Severity of Alternaria Leaf Blotch Disease of the Apple Tree. *Remote Sensing*, 14(11), p.2519. Available from: https://doi.org/10.3390/rs14112519.

[19] Liu, Y., Liu, J., Cheng, W., Chen, Z., Zhou, J., Cheng, H. and Lv, C., 2023. A High-Precision Plant Disease Detection Method Based on a Dynamic Pruning Gate Friendly to Low-Computing Platforms. *Plants*, 12(11), p.2073. Available from: https://doi.org/10.3390/plants12112073.

[20] Mamun, S.S., Ren, S., Rakib, M.Y.K. and Asafa, G.F., 2025. WGA-SWIN: Efficient Multi-View 3D Object Reconstruction Using Window Grouping Attention in Swin Transformer. *Electronics*, 14(8), p.1619. Available from: https://doi.org/10.3390/electronics14081619.

[21] Nazir, T., Iqbal, M.M., Jabbar, S., Hussain, A. and Albathan, M., 2023. EfficientPNet—An Optimized and Efficient Deep Learning Approach for Classifying Disease of Potato Plant Leaves. *Agriculture*, 13(4), p.841. Available from: https://doi.org/10.3390/agriculture13040841.

[22] Ojo, M.O. and Zahid, A., 2023. Improving Deep Learning Classifiers Performance via Preprocessing and Class Imbalance Approaches in a Plant Disease Detection Pipeline. *Agronomy*, 13(3), p.887. Available from: https://doi.org/10.3390/agronomy13030887.

[23] Orchi, H., Sadik, M., Khaldoun, M. and Sabir, E., 2023. Automation of Crop Disease Detection through Conventional Machine Learning and Deep Transfer Learning Approaches. *Agriculture*, 13(2), p.352. Available from: https://doi.org/10.3390/agriculture13020352.

[24] Padshetty, S. and Ambika, 2023. Leaky ReLU-ResNet for Plant Leaf Disease Detection: A Deep Learning Approach. *Engineering Proceedings*, 59(1), p.39. Available from: https://doi.org/10.3390/engproc2023059039.

[25] Parez, S., Dilshad, N., Alghamdi, N.S., Alanazi, T.M. and Lee, J.W., 2023. Visual Intelligence in Precision Agriculture: Exploring Plant Disease Detection via Efficient Vision Transformers. *Sensors*, 23(15), p.6949. Available from: https://doi.org/10.3390/s23156949.

[26] Phinzi, K., Abriha, D. and Szabó, S., 2021. Classification Efficacy Using K-Fold Cross-Validation and Bootstrapping Resampling Techniques on the Example of Mapping Complex Gully Systems. *Remote Sensing*, 13(15), p.2980. Available from: https://doi.org/10.3390/rs13152980.

[27] Pineda Medina, D., Miranda Cabrera, I., Cruz, R.A. de la, Guerra Arzuaga, L., Cuello Portal, S. and Bianchini, M., 2024. A Mobile App for Detecting Potato Crop Diseases. *Journal of Imaging*, 10(2), p.47. Available from: https://doi.org/10.3390/jimaging10020047.

[28] Saleem, M.H., Potgieter, J. and Arif, K.M., 2020. Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers. *Plants*, 9(10), p.1319. Available from: https://doi.org/10.3390/plants9101319.

[29] Saleem, S., Sharif, M.I., Sharif, M.I., Sajid, M.Z. and Marinello, F., 2024. Comparison of Deep Learning Models for Multi-Crop Leaf Disease Detection with Enhanced Vegetative Feature Isolation and Definition of a New Hybrid Architecture. *Agronomy*, 14(10), p.2230. Available from: https://doi.org/10.3390/agronomy14102230.

[30] Shah, S., Taj, I., Usman, S., Shah, S., Imran, A. and Khalid, S., 2024. A hybrid approach of vision transformers and CNNs for detection of ulcerative colitis. *Scientific Reports*, 14, 10. Available from: https://doi.org/10.1038/s41598-024-75901-4.

[31] Shah, S.R., Qadri, S., Bibi, H., Shah, S.M.W., Sharif, M.I. and Marinello, F., 2023. Comparing Inception V3, VGG 16, VGG 19, CNN, and ResNet 50: A Case Study on Early Detection of a Rice Disease. *Agronomy*, 13(6), p.1633. Available from: https://doi.org/10.3390/agronomy13061633.

[32] Sun, Y., Ning, L., Zhao, B. and Yan, J., 2024. Tomato Leaf Disease Classification by Combining EfficientNetv2 and a Swin Transformer. *Applied Sciences*, 14(17), p.7472. Available from: https://doi.org/10.3390/app14177472.

[33] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A. and Jegou, H.,

2021. Training data-efficient image transformers & distillation through attention. In: M. Meila and T. Zhang, eds. *Proceedings of the 38th International Conference on Machine Learning*, *Proceedings of machine learning research*, vol. 139. PMLR, pp.10347–10357. Available from: https://proceedings.mlr.press/v139/touvron21a.html.

[34] Yu, D., Wan, B. and Sheng, Q., 2024. Automated Generation of Urban Spatial Structures Based on Stable Diffusion and CoAtNet Models. *Buildings*, 14(12), p.3720. Available from: https://doi.org/10.3390/buildings14123720.

[35] Zhang, Z.Y., Yan, C.X., Min, Q.M., Zhang, Y.X., Jing, W.F., Hou, W.X. and Pan, K.Y., 2024. Leverage Effective Deep Learning Searching Method for Forensic Age Estimation. *Bioengineering*, 11(7), p.674. Available from: https://doi.org/10.3390/bioengineering11070674.

[36] Zhu, D., Tan, J., Wu, C., Yung, K. and Ip, A.W.H., 2023. Crop Disease Identification by Fusing Multiscale Convolution and Vision Transformer. *Sensors*, 23(13), p.6015. Available from: https://doi.org/10.3390/s23136015.