Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Novel Hardware-Aware Multi-Objective Evolutionary Neural Architecture for Crop-Disease Classification on Edge Devices

Anitha K
Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya
Naveenkumar R
drnk1983@gmail.com

CGC University Mohali Punjab India
Pooja H L
Ramaiah Institute of Technology
Shantha Kumar M
Sree Sakthi Engineering College, Karamadai,
Valarmathi K
S.A. Engineering College
Monica K M
School of Computer Science and engineering. VIT,

Research Article

Keywords: Crop Disease Detection, EfficientNets, Non-dominated Sorting Genetic Algorithm II (NSGA-II), Edge Devices

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11351948/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A NOVEL HARDWARE-AWARE MULTI-OBJECTIVE EVOLUTIONARY NEURAL ARCHITECTURE FOR CROP-DISEASE CLASSIFICATION ON EDGE DEVICES.

1. Anitha K
Assistant Professor
Department of ECE, SCSVMV University
anithakece@kanchiuniv.ac.in
ORCID id: https://orcid.org/0009-0003-8347-9554

2. Naveenkumar. R*
CGC University Mohali, Dept of CSE, Punjab,140307 – India.
drnk1983@gmail.com
ORCID id: https://orcid.org/0000-0001-9033-9400
*Corresponding Author

3.Pooja H L
Assistant Professor
Department of Electronics and Telecommunication Engineering
M S Ramaiah Institute of Technology, Bengaluru, Karnataka
poojahl@msrit.edu
ORCID id: https://orcid.org/0009-0004-6953-9263

4.Shantha Kumar M
Associate Professor,
Department of Electronics and Communication Engineering,
Sree Sakthi Engineering College, Karamadai, Coimbatore.
mpshanthakumar@gmail.com
ORCID id: https://orcid.org/0000-0002-8632-3067

5. Valarmathi K
Assistant Professor
Department of Information Technology
S.A. Engineering College
valarmathik@saec.ac.in
ORCID id: https://orcid.org/0009-0007-3461-5424

6.Monica K M
Assistant professor
School of Computer Science and engineering. VIT,
Chennai.
monica.km@vit.ac.in

Abstract

Crop disease detection plays a pivotal role in improving yields and quality, thereby supporting the efficient utilization of agricultural resources. Conventional detection methods suffer from limited scalability, subjective detection, and reduced reliability under varying environmental conditions. Machine learning and deep learning-based approaches have demonstrated their strong capabilities for efficient crop disease classification from versatile field and plant images. However, non-scalable architectures of these existing models consume large amounts of parameters, computational overhead, memory, and high latency. To address the aforementioned challenge, this research paper proposes hardware-aware, multi-objective, evolutionary search-based neural architectures for efficient crop detection on low-power edge

devices. The proposed framework utilizes an EfficientNet-Lite model as the main component, whose parameters are optimized by the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to achieve high classification performance, low inference latency, and limited computational parameters. In the proposed optimization, the framework predicts the performance of learning networks and estimates their execution time on the target edge hardware without testing each model directly. It then identifies a set of efficient architectures with varying accuracy, latency, and energy trade-offs, enabling the selection of the most suitable model based on the target device's requirements. Extensive Experimentation is carried out using PlantVillage datasets, and performance metrics such as accuracy, precision, recall, F1-score, latency, and computational parameters are calculated. Results illustrates the suggested framework has achieved the accuracy of 0.98, precision of 0.9734, recall of 0.973 with latency and parameters on different edge architectures. Compared with EfficientNet-B0 and MobileNetV3, it is [A]× faster and [B]× smaller, with accuracy within [C]%. These results show that hardware-aware evolutionary search can produce compact and accurate models for real-time plant disease monitoring in the field

Keywords: Crop Disease Detection, EfficientNets, Non-dominated Sorting Genetic Algorithm II (NSGA-II), Edge Devices.

1. Introduction

Agriculture plays a pivotal role in improving the country's economic development by ensuring food security and sustainable production of yields. With a growing population and unpredictable climatic conditions, precision agriculture has emerged as an important approach to improving crop productivity through data-driven monitoring and intelligent decision support systems. Precision Agriculture has increased profitability by an average of 18.5%, reduced resource input costs by 8% to 20%, and boosted the country's economic development [1,2]. In recent times, Plant diseases have adverse effects on yield production and remain a major challenge among farmers and researchers. Hence, early detection enables farmers to implement appropriate preventive practices before disease progression becomes severe, thereby reducing crop losses and unnecessary pesticide usage[3,4].

In recent years, Machine Learning (ML) and Deep Learning (DL) have played a significant role in detecting plant diseases by analyzing textures and other visual patterns. Among them, Deep learning algorithms have gained more popularity in designing intelligent systems by learning discriminative features directly from plant images. Convolutional Neural Networks (CNNs), Vision Transformers, and Hybrid architectures can therefore provide highly accurate disease classification across multiple crops and disease categories[6,7]. Also, lightweight architectures such as MobileNet, ShuffleNet, and EfficientNet are deployed to improve the trade-off between recognition accuracy and computational efficiency[8,9]. However, high-performing DL models can contain millions of parameters and require substantial computational resources, memory, and inference time. This creates a challenge for deployment on resource-constrained agricultural edge devices. Hence the strong optimization technique is required for automatically tuning the parameters of DL to enhance the disease classification performance and hardware-related objectives such as computational complexity, memory consumption, inference latency, and energy usage.

1.1 Motivation and Research Contribution

Motivated by the aforementioned challenge, this research paper proposes a novel hardware–aware architecture that uses EfficientNetV2 as the backbone, with the Non-dominating Sorting Genetic Algorithm-II(NSGA-II) employed to optimize architectural parameters that balance classification performance and computational overhead. The key contribution of the research is as follows

1. The paper proposes a hardware-aware architecture in which a lightweight EfficientNetV2 is employed to optimize architectural parameters, thereby enhancing performance and reducing computation.
2. The proposed framework incorporates Non-Dominated Sorting Genetic Algorithm II (NSGA-II) to simultaneously optimize disease classification performance and hardware deployment constraints.
3. Extensive experimentation is carried out using PlantVillage Datasets and various performance metrics are measured by deploying on the different edge architectures. Experimental results demonstrate that the proposed optimized architecture shows significantly better performance than other existing lightweight architectures.

1.2 Organization of the Paper

The remainder of this paper is structured as follows: Section-2 provides comprehensive insights into the ML/DL frameworks utilized for plant disease detection. The workflow of the suggested model is presented in Section 3. The proposed framework is experimentally evaluated together with its comparative analysis in Section-4. Finally, future research directions are discussed in Section-5.

2. Related Works

Ye et al. [10] proposed PestNAS, a field pest detection network that couples adaptive feature fusion with evolutionary neural architecture search.The model focuses on the use of plant context information and the reliance on hand-designed architectures. The model replaces gradient-based search, which tends to settle in optima with a search space that evolves through statistical operations and adaptive feature-weight mutation. In experiments on eleven field pests, PestNAS—the model—achieved higher detection accuracy, than ten other architecture search models. Nevertheless, the work addresses pest detection rather than leaf disease classification, so its transfer to disease recognition remains unexamined.

Omole et al. [11] introduced AgriNAS, which integrates a neural architecture search framework with an adaptive convolutional architecture for soybean pest and disease detection. The framework combines a Spatial–Time Augmentation strategy, a multi-stage convolutional network that adapts to input complexity, bi-level optimization and entropy-based regularization, and it classified more accurately than VGG-19 and a transfer-learning CNN. However, the search and training rely on substantial GPU resources, and the framework was evaluated on a three-class soybean dataset covering caterpillar damage, Diabrotica speciosa damage and healthy leaves.

Saleem et al. [12] proposed MUTPSO-CNN, which applies a mutant particle swarm optimization algorithm to search for an optimum CNN architecture for paddy leaf disease classification. The authors motivated the work by the dependence of hand-built CNNs on expert knowledge and heavy computation, and reported that the searched architecture outperformed hand-crafted CNNs in accuracy, precision/recall and execution time. Nevertheless, the study is confined to paddy leaf disease classification, and the generalization of the searched architecture to other crops is not examined.

Birajdar et al. [13] introduced a CNN made up of three convolutional blocks using 32, 64 and 128 filters, followed by two fully connected layers. This model was designed for real-time crop disease detection. It was trained on a subset of 1,000 images from the PlantVillage dataset focusing on tomato, potato and pepper leaves. The training included image augmentation and careful tuning of hyperparameters. The model achieved a test accuracy of 90.5 percent. Its file size is 3.6 MB and it processes each image in about 78 milliseconds on a Raspberry Pi 4. The authors note that the images were taken under controlled conditions with backgrounds. They also admit that performance might not hold up well in complex outdoor settings. Additionally their approach only classifies leaves healthy or diseased—no further disease types are identified.

Kalaivani et al. [14] proposed a model called C-CROP for detecting crop diseases. This model combines a CNN for feature extraction from leaf images with an SVM to perform classification. The CNN eliminates the need to manually design features while the SVM works to define decision boundaries based on those features. When tested on a dataset of crop disease images the model hit an accuracy of 95.2 percent and performed better than traditional methods. There is no mention of inference time for the model runs on hardware with resources. Also both the CNN and SVM parts were handcrafted of being discovered through automated architecture search. As a result the model cannot adjust its structure when new data comes in.

Slimani et al. [15] proposed a learning model using neural architecture search to monitor growth stages of Vicia faba L. From drone footage. The model was evaluated against seven models using 2,530 UAV images captured under different lighting and background conditions. It reached 95.80 percent precision, 98.80 percent recall and 71.30 percent mAP@0.5:0.95. It showed a balance between precision, speed and model size. The results suggest it could be useful for real-time monitoring, in applications.. Nevertheless, the work addresses growth-stage recognition rather than disease identification, and it is limited to a single crop.

Uchitkar et al. [16] developed a plant disease detection system that uses a CNN with pooling, dropout and dense layers to classify leaf images. The plant disease detection system was trained on, than 70,000 images covering 38 classes, including 26 disease categories and 12 healthy types. Data augmentation, preprocessing and hyperparameter tuning were used to improve generalization. The plant disease detection system reached a validation accuracy of 98.04 percent. It was then deployed in a Flask web application that returns real-time diagnoses from leaf images.Nevertheless, the reported accuracy is a validation figure, and the architecture is manually designed rather than searched. The authors themselves list dataset expansion, ensemble learning and mobile platform support as future improvements.

Barbedo [17] investigated how the size and variety of image datasets affect the effectiveness of deep learning in plant pathology, using a database of 12 plant species that differ in sample count, number of diseases and range of conditions. The results indicate that the technical constraints of automatic plant disease classification have largely been overcome. However, training on limited image datasets still has undesirable consequences that prevent the wider adoption of the technology. As an empirical analysis rather than a new method, the study identifies data dependence as the central obstacle but does not propose a way to reduce it.

Nawani et al. [18] carried out an empirical performance analysis of modified convolutional neural network models for edible plant leaf disease detection, examining how adjustments to standard CNN architectures affect classification performance. Nevertheless, performance gains obtained by modifying a fixed CNN design depend on the particular modifications chosen and the leaf datasets used, and they do not by themselves provide a systematic way to find an architecture suited to a new dataset.

Slimani et al. [19] applied neural architecture search within deep transfer learning to identify the degree of rust infection in Vicia faba L. Ten deep learning models were assessed through transfer learning, and all reached F1 scores above 90.0%, with the search-based model performing best at 90.84%. Nevertheless, the study concerns a single disease on a single crop, and other models achieved F1 scores close to 90.0%, so the advantage of the searched architecture is modest.

Abouelmagd et al. [19] proposed an optimized capsule neural network to detect and classify ten tomato leaf diseases. The approach was motivated by the limited ability of CNNs to capture relative spatial and orientation relationships, and augmentation and preprocessing were used to reduce overfitting. On a dataset of 70,834 images, the capsule network reached 96.39% accuracy against 92.87% for a conventional CNN. Nevertheless, class-wise results were uneven, with accuracy below 60% for one of the ten classes, and the study is limited to a single crop.

Table 1. Summary of related work on neural architecture search and CNN-based crop disease detection.

Author	**Key Findings**	**Pros**	**Cons**
Ye et al.	PestNAS (adaptive feature fusion with evolutionary NAS) achieved higher accuracy than ten other NAS models on eleven field pests.	Automates architecture design; avoids the local optima of gradient-based search; uses plant context.	Pest detection only; transfer to disease classification not examined.
Omole et al.	AgriNAS (NAS with adaptive convolution and Spatial–Time Augmentation) outperformed VGG-19 and a transfer-learning CNN on soybean.	Adapts to input complexity; regularized against overfitting.	High GPU demand; three-class soybean dataset.

Author	Key Findings	Pros	Cons
Saleem et al.	MUTPSO-CNN found architectures that beat hand-crafted CNNs in accuracy, precision/recall and execution time.	Reduces dependence on expert design; automated search.	Paddy only; generalization to other crops not examined.
Birajdar et al.	Lightweight CNN reached 90.5% accuracy, 3.6 MB size and 78 ms latency on a Raspberry Pi 4.	Compact; real-time on edge devices.	Controlled images; binary classification only; may degrade in field scenes.
Kalaivani et al.	C-CROP (CNN with SVM) achieved 95.2% accuracy.	No manual feature design; SVM handles complex boundaries.	Limited dataset detail; no inference or hardware results; fixed hand-designed structure.
Slimani et al.	NAS model for UAV growth-stage monitoring reached 95.80% precision, 98.80% recall and 71.30% mAP@0.5:0.95.	Balances precision, speed and model size.	Growth stages, not disease; single crop.
Uchitkar et al.	CNN trained on over 70,000 images (38 classes) reached 98.04% validation accuracy; deployed as a Flask web app.	High accuracy; real-time web diagnosis.	Validation accuracy only; manually designed architecture.
Barbedo	Dataset size and variety strongly affect deep learning across 12 plant species; limited data hinders adoption.	Identifies data dependence as the key obstacle.	Proposes no method to reduce it.
Nawani et al.	Empirical analysis of modified CNNs for edible plant leaf disease detection.	Compares modified CNN variants empirically.	Manual modification; no systematic architecture search.
Slimani et al.	NAS-based model gave the best F1 (90.84%) among ten transfer-learning models for rust infection levels.	Identifies infection degree; uses transfer learning.	Single disease and crop; narrow margin over other models.
Abouelmagd et al.	Capsule network reached 96.39% accuracy versus	Captures spatial and orientation relationships.	Uneven class accuracy (below 60% for one class); single crop.

Author	Key Findings	Pros	Cons
	92.87% for a CNN on ten tomato diseases.		

3.Proposed Framework

Figure 1 illustrates the proposed workflow for hardware-aware plant disease detection. The framework consist of Image Collection, Preprocessing, Optimized Model and Performance evaluation. Initially, PlantVillage image are first subjected to image enhancement, normalization, and GAN-based augmentation. The resulting images are processed using a lightweight EfficientNetV2 backbone comprising multiple feature extraction stages, followed by a classification head for disease prediction. NSGA-II is subsequently employed to optimize the network architecture according to classification performance and computational constraints. The selected Pareto-optimal architecture is deployed on resource-constrained edge platforms and evaluated using both classification and deployment-oriented performance metrics. The detailed description of each component is as follows:

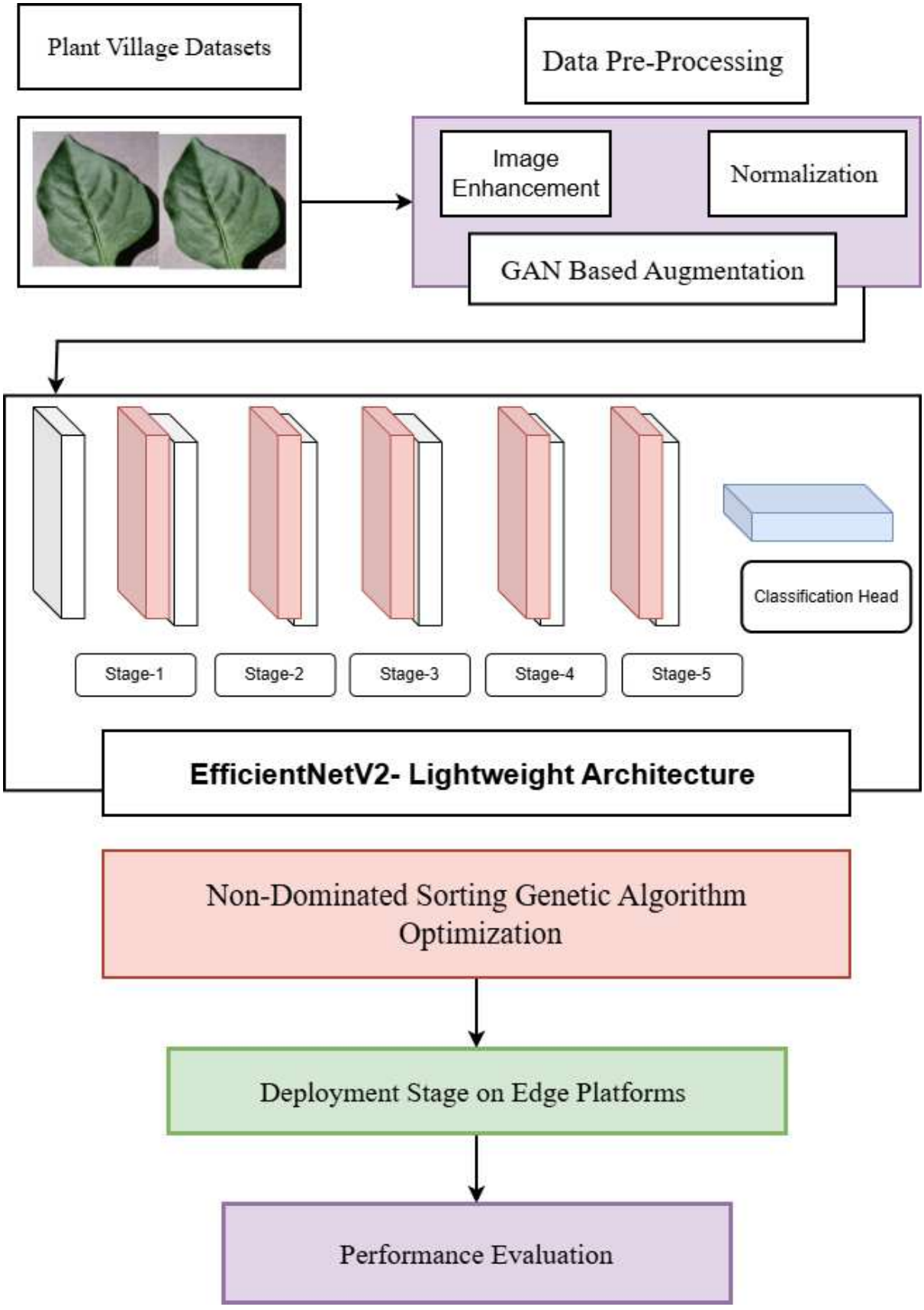


Figure 1 Overall Architecture for the Proposed NSGA-II based EfficientNetV2 Framework

3.1 PlantVillage Dataset Description

In this research, PlantVillage datasets obtained from source[20] has been used for training and testing. PlantVillage dataset have 82,286 images belonging to 14 different plants. There are total 38 classes, out of which 26 depicts diseases from various plants, whereas 12 are classes of different plant with healthy leaves which is illustrated in Figure 2 (a-b) and Figure 3 respectively.

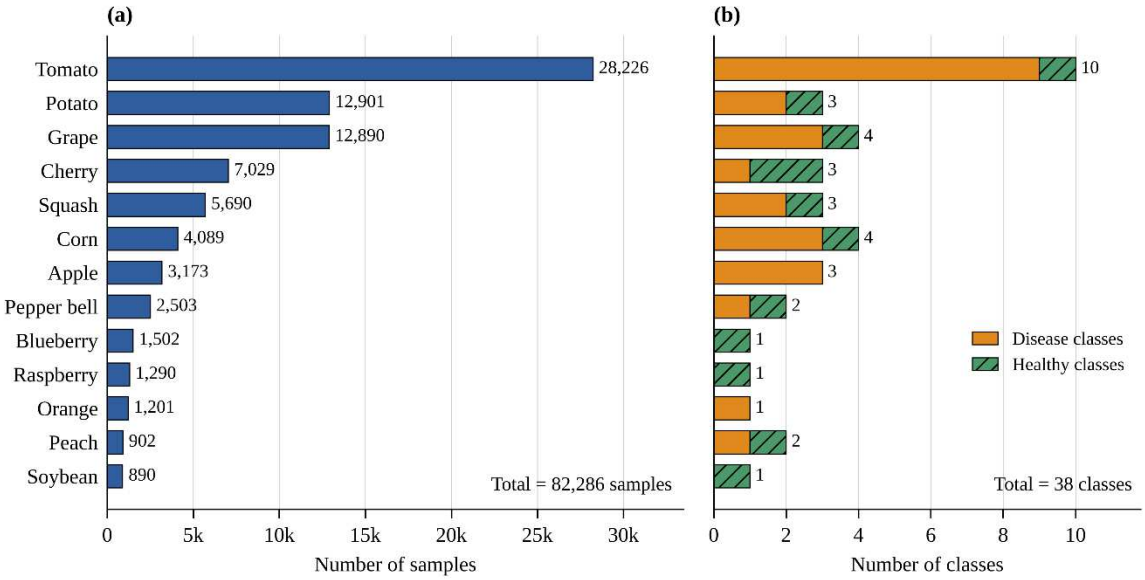


Figure 2 Class Distribution Analysis a) Total Number of Samples b) Total Number of Classes



(a)



(b)



(c)



(d)

Figure 3 Visual Representation of Healthy and Disease Sample Plant a) Healthy Apple b) Pepper bell-bacterial Spot c) Apple Black rot d) Tomato Diseases.

3.2 Data Preprocessing

In this research, CLAHE (Contrast Limited Adaptive Histogram Equalization) is used as the most promising pre-processing technique to remove the noise levels from the images and enhancing the contrast on the images which aid for better classification. Following the pre-processing technique, image augmentation process is adopted to increase the sign images that suits for an effective training without any over-fitting problem. In this framework, the GAN technique [21] is adopted as the augmentation technique to overcome the problem of overfitting. Figure 4 presents the class distribution after applying the augmentation technique. Table 2 presents the complete datasets description used for training (70%), testing (15%) and validation (15%) respectively.

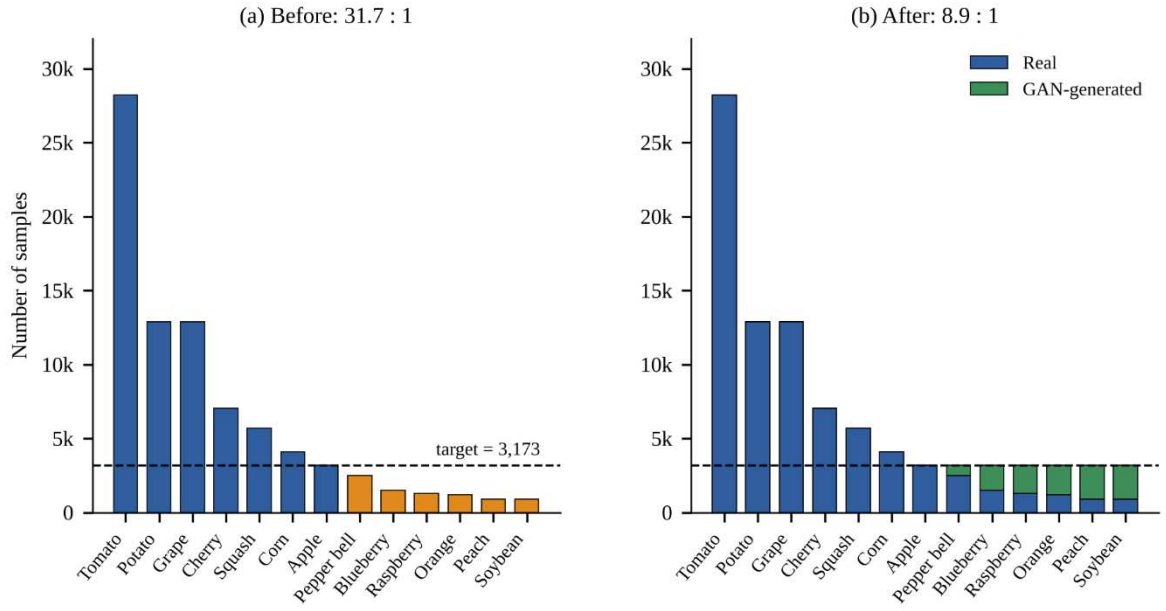


Figure 4 Class Distribution Process a) Before Applying GAN b) After Applying GAN Model

Table 2 Training/Testing/Validation Process for the Experimentation

Description	Before GAN	After GAN
Total number of Training Data	57,600	65,125
Testing data	12,343	12,433
Validation	12,343	12,343
Total Number of data	**82,286**	**89,811**

3.3 EfficientNetV2 Architecture

EfficientNetV2 is a category of convolutional neural networks designed to achieve efficient performance and less computational complexity. The architecture primarily employs Fused MBConv and MBConv blocks to extract hierarchical representations from input images. Fused-MBConv blocks combine convolutional operations within the early stages of the network, enabling efficient feature extraction from high-resolution input representations, whereas MBConv blocks employ depth-wise convolution and expansion–projection operations to reduce computational cost in deeper stages. The architecture also incorporates Squeeze-and-Excitation (SE) mechanisms, which adaptively optimize channel-wise feature responses and enhance the most informative disease-related patterns. In the proposed framework, pre-processed input images are passed through the EfficientNetV2 stages, where shallow layers capture low-level characteristics such as edges, color variations, and texture patterns, while deeper layers learn high-level patterns associated with disease symptoms, including lesions, spots, discoloration, and morphological changes. The extracted feature representation is subsequently aggregated using global average pooling and passed to a classification layer to predict the corresponding crop-disease category. Figure 5 presents the EfficientNetV2 layers used for the experimentation process.

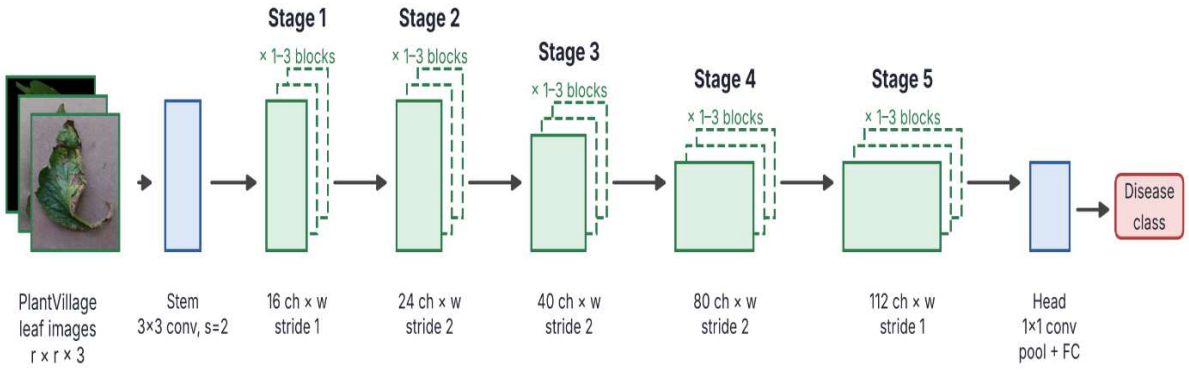


Figure 5 EfficientNetV2 – Architectural Representation illustrating the variable-depth feature extraction stages and classification head.

As shown in Figure 5, pre-processed image is initially processed using a 3x3 convolutional stem with a stride of two to extract low-level visual features while reducing the spatial resolution. The resulting representation is subsequently propagated through five feature-extraction stages with channel dimensions of 16, 24, 40, 80, and 112, respectively. Each stage contains a variable number of computational blocks, ranging from one to three, allowing the neural architecture search process to optimize the depth of individual stages according to the target objectives. . Following the final feature-extraction stage, 1x1 convolutional head, global pooling operation, and fully connected classifier transform the learned representation into the corresponding crop-disease class. The variable stage depth, together with other architectural parameters, forms the search space explored by proposed evolutionary to identify architectures that achieve an appropriate trade-off among disease classification performance, computational complexity, inference latency, memory usage, and energy consumption. Figure 6 presents the detailed description of each stages which is used as search space of the proposed model.

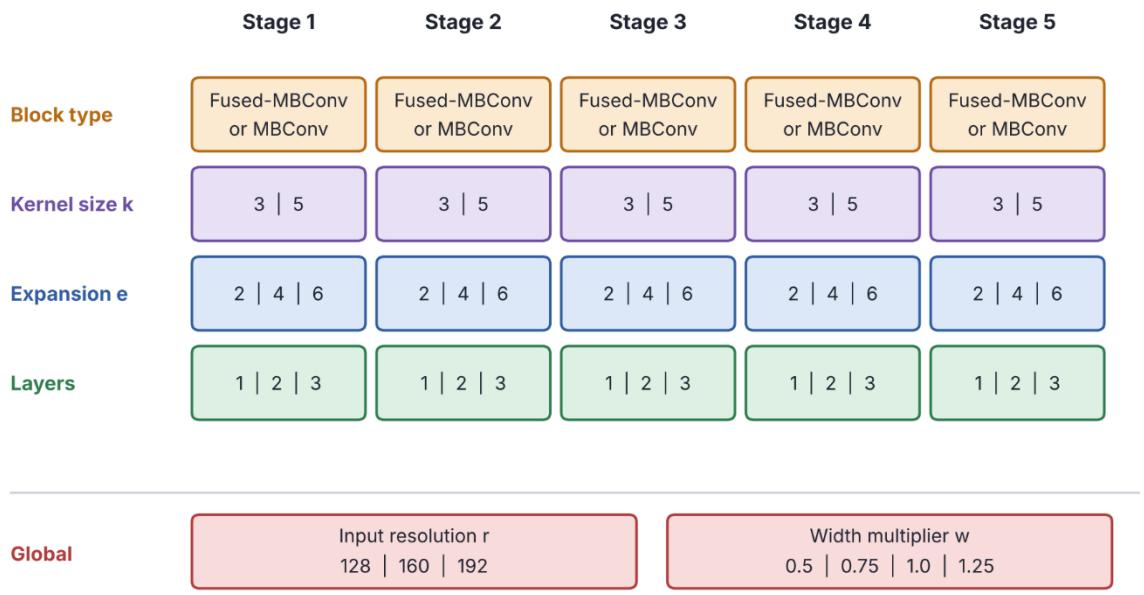


Figure 6 Searchable Space of the EfficientNetV2 architecture to generate Surrogate Predictor for efficient classification of plant diseases

For each of the five feature-extraction stages, the search process determines the block type between Fused-MBConv and MBConv, the kernel size between 3x3 and 5x5, the expansion ratio among 2, 4, and 6, and the number of repeated blocks between 1 and 3. In addition, two global parameters, namely the input resolution (128, 160, 192) and the width multiplier (0.5, 0.75, 1.0, 1.25), are jointly optimized. These parameters control the representational capacity and computational requirements of each candidate architecture. The resulting search space contains approximately 700 million possible configurations, making exhaustive evaluation computationally impractical. Therefore, NSGA-II efficiently explores the architecture space by generating, evaluating, selecting, recombining, and mutating candidate networks according to classification performance and hardware-related objectives. The final Pareto-optimal

architectures offer different trade-offs among disease-detection performance, computational complexity, inference latency, memory consumption, and parameters.

3.4 NSGA-II Based EfficientNetV2 Architecture

The proposed architecture employs a metaheuristic optimization technique from the genetic algorithm category. These algorithms have demonstrated their effectiveness in finding the optimal solutions in non-differentiable spaces. Furthermore, Pareto-based multi-objective evolutionary algorithms are used to optimize network parameters based on Pareto efficiency.

3.4.1 Problem Formulation

Let A denote the pre-trained EfficientNetV2 model with desired performance metrics. Also, let the parameters of the pre-trained model be denoted by. Further, N measures hardware parameters, including computational performance and latency. The first two genes select the resolution and width multiplier. Each stage then contributes four genes: block type (b), kernel size (k), expansion ratio (e), and depth of layers (d). Every gene stores an index into its list of options, so every chromosome decodes to a valid network.

$$x = [r,\ w,\ (b_1, k_1, e_1, d_1),\ (b_2, k_2, e_2, d_2),\ \ldots,\ (b_S, k_S, e_S, d_S)] \quad (1)$$

Hence, the multi-objective hardware-aware search problem can be formulated as

$$f_1(x) = 1 - \mathrm{Acc}_{val}\big(N(x)\big), \quad f_2(x) = \mathrm{Lat}_H\big(N(x)\big), \quad f_3(x) = \mathrm{Params}\big(N(x)\big) \quad (2)$$

Here, $f1$ is the validation error after a short training run, $f2$ is the inference latency in milliseconds on the target hardware H, and f_3 is the parameter count in millions.

3.4.2 Searching Space Mechanism

Initially, a population of candidate architectures is generated by varying architectural parameters and treated as 'R' random chromosomes. Each chromosome is decoded into a network and trained for a few epochs on a stratified subset of pre-processed image datasets. Each chromosome is evaluated using the multi-objective function given in Equation(2). The non-dominating sorting is then applied to classify these parameters into distinct Pareto fronts, while the crowding distance preserves diversity among the participating populations, thereby preventing premature convergence toward similar network configurations and maintaining diverse accuracy, latency, and parameter trade-offs. The crowding distance for i solution can be calculated as follows

$$CD_i = \sum_{m=1}^{3} \frac{f_m^{(i+1)} - f_m^{(i-1)}}{f_m^{max} - f_m^{min}} \quad (3)$$

The high Pareto solutions are selected as parents and subjected to crossover and mutation operations to generate new candidate configurations. The newly generated architectures are subsequently evaluated and combined with the existing population to form the next generation.

This evolutionary process is repeated over successive generations, progressively exploring promising regions of the EfficientNetV2 search space. Algorithm 1 presents the complete working algorithm for the proposed search mechanism.

Algorithm 1: Hardware-aware NSGA-II NAS for crop disease detection

```
Input: dataset D, population size N, generations G, target device H
Output: Pareto front P*, deployed model
 1: P_0 <- N random chromosomes
 2: for t = 0 to G-1 do
 3:     for each chromosome x in P_t (not cached) do
 4:         N(x) <- decode(x)
 5:         train N(x)  on a subset of D
 6:         f1 <- 1 - validation accuracy
 7:         f2 <- latency of N(x) on H
 8:         f3 <- parameters of N(x)
 9:     end for
10:     rank P_t by non-dominated sorting; compute crowding distance
11:     Q_t <- crossover and mutation of parents chosen by binary tournament
12:     evaluate Q_t as in lines 3-9
13:     R_t <- P_t U Q_t
14:     P_(t+1) <- best N of R_t by front rank, then crowding distance
15: end for
16: P* <- first front of P_G
17: select x* from P* under the device budget
18: retrain N(x*) on D; quantize to INT8; deploy on H
19: return P*, N(x*)
```

To reduce computational cost, the proposed framework can employ a surrogate predictor to estimate classification performance and a hardware-specific latency lookup table to estimate execution time. Finally, the search produces a Pareto-optimal set of EfficientNetV2 architectures, from which a deployment architecture can be selected according to the objective function as mentioned in Equation(2). Table 3 presents the optimized parameters of NSGA-II –EfficientNets for crop-disease detection.

Table 3. Layer-wise configuration of the final architecture selected by NSGA-II (r = 160, w = 1.0)

Stage	Block type	Kernel (k)	Expand (e)	Repeats	Stride	Output channels	Output size
Input	RGB leaf image	–	–	–	–	3	160×160
Stem	Conv 3×3 + BN + SiLU	3	–	1	2	24	80×80
1	Fused-MBConv	3	4	1	1	16	80×80
2	Fused-MBConv	3	4	2	2	24	40×40
3	MBConv + SE	5	4	2	2	40	20×20

4	MBConv + SE	3	6	2	2	80	10×10
5	MBConv + SE	5	6	1	1	112	10×10
Head	Conv 1×1 + global pool + FC	1	–	1	1	320 → classes	1×1

4. Results and Discussion

The proposed learning approach is developed in Python 3.11, with the use of the PyTorch 2.1 on CUDA12.1. All the experiments were run on a workstation with an AMD Ryzen 9 5900X (12-core, 4.8 GHz) processor and an NVIDIA GeForce RTX 3090 GPU with 24 GB VRAM, providing enough computational power for training. The proposed framework is depoloyed on the different edge processors such as Raspberry pi 4 and NVIDIA Jetson Nano boards. Table 4 presents the simulation parameters used in the experiments.

Table 4 Simulation parameters used for experimentation

Group	Parameter	Value
Search space	Input resolution r	{128, 160, 192}
	Width multiplier w	{0.5, 0.75, 1.0, 1.25}
	Number of stages	5 (22-gene integer chromosome)
	Block type (per stage)	{Fused-MBConv, MBConv + SE}
	Kernel size k	{3, 5}
	Expansion ratio e	{2, 4, 6}
	Depth (repeats per stage)	{1, 2, 3}
NSGA-II	Population size	12
	Generations	10
	Objectives (all minimised)	Validation error, CPU latency (ms), parameters (M)
	Selection	Binary tournament (rank, then crowding distance)
	Crossover	SBX, probability 0.9, $\eta = 15$
	Mutation	Polynomial, $\eta = 20$
	Repair / duplicates	Rounding repair; duplicates eliminated
	Random seed	1
Candidate training	Epochs per candidate	3
	Optimiser	AdamW, learning rate 2×10^{-3}, weight decay 10^{-4}
	Loss / batch size	Cross-entropy / 64
	Input normalisation	ImageNet mean and standard deviation

	Latency measurement	Batch size 1, CPU, 5 warm-up + 20 timed runs
GAN	Model	Conditional WGAN-GP, image size 64×64
	Latent dimension / base width	128 / 64
	Epochs / batch size	200 / 64
	Critic steps per generator step	5
	Gradient-penalty weight λ	10
	Optimiser	Adam, learning rate 10^{-4}, β = (0, 0.9)
	Oversampling target	Median class size (training split)

For evaluating the performance of the proposed framework, five standard performance parameters are considered, which are shown in Table 5 These results form the basis of calculating accuracy, precision, recall, and F1-Score, which can be used to evaluate the performance of the proposed approach in all disease categories

Table 5 Measures of performance adopted for the assessment

SL.NO	Performance Measures	Expression
1	Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
2	Recall	$\frac{\text{TP}}{\text{T P+FN}}$ x100
3	Precision	$\frac{TN}{TP + FP}$
4	F1-Score	$2.\frac{Precision * Recall}{Precision + Recall}$

TP and TN denote True Positive and True Negative, respectively, whereas FP and FN represent False Positive and False Negative, respectively.

4.1 Result Outcome Analysis

Table presents the performance metrics of the proposed surrogate framework in detecting diseases from different plants. Furthermore, performance has been demonstrated on different edge platforms, such as the Raspberry Pi Model 4B and the NVIDIA JETSON Nano board. From the Table 6, it is evident that the proposed model has achieved promising performance on the CPU workstation. Table also demonstrates that the efficiency of the NSGA-II-optimized EfficientNetV2 has also achieved the significant performance in the edge architectures.

Table 6 Performance Metrics of the Proposed NGSA-II Optimized EfficientNetV2 Nets in CPU Workstation.

Plant	Type	Performance Metrics				
		Accuracy	Precision	Recall	F1-score	AUC
Tomato	Healthy	0.9832± 0.00143	0.9622± 0.0012	0.9601± 0.0022	0.960± 0.00112	0.980

	Disease	0.9820± 0.00112	0.9653± 0.00116	0.9563± 0.0011	0.9503± 0.0018	0.972
Apple	Healthy	0.9783± 0.00102	0.978± 0.00152	0.9618± 0.00121	0.973± 0.00112	0.980
	Disease	0.9803± 0.0018	0.9802± 0.00151	0.9701± 0.00109	0.9702± 0.00108	0.9873
Blueberry	Healthy	0.9805± 0.00152	0.973± 0.00109	0.968± 0.00112	0.9609± 0.0011	0.9864
	Disease	0.986± 0.00103	0.982± 0.00110	0.978± 0.00101	0.967± 0.0011	0.993
Orange	Healthy	0.9800± 0.00119	0.9812± 0.0015	0.972± 0.0011	0.969± 0.0009	0.983
	Disease	0.981± 0.00117	0.9806± 0.00112	0.963± 0.0016	0.973± 0.0015	0.9783
Pepperbell	Healthy	0.9812± 0.00112	0.978± 0.00118	0.963± 0.00110	0.9642± 0.00110	0.9833
	Disease	0.9810± 0.00114	0.9753± 0.00115	0.9742± 0.00112	0.9673± 0.00110	0.9673
Potato	Healthy	0.9753± 0.00102	0.967± 0.00112	0.975± 0.00114	0.9832± 0.0012	0.9833
	Disease	0.9703± 0.0012	0.9653± 0.00102	0.9723± 0.00101	0.9645± 0.00129	0.9845
Corn	Healthy	0.9705± 0.0011	0.9654± 0.00110	0.9742± 0.00102	0.9653± 0.00102	0.9844
	Disease	0.9756± 0.00112	0.9673± 0.00112	0.9853± 0.00101	0.9753± 0.00110	0.9903
Raspberry	Healthy	0.9753± 0.00162	0.9653± 0.00161	0.962± 0.00110	0.967± 0.00118	0.9902
	Disease	0.9752± 0.00102	0.9742± 0.0011	0.962± 0.00102	0.9653± 0.00102	0.9880
Soyabeans	Healthy	0.9786± 0.00112	0.9742± 0.00102	0.973± 0.0011	0.973± 0.0010	0.9887
	Disease	0.9867± 0.00102	0.9752± 0.00112	0.975± 0.00103	0.9653± 0.00102	0.9902
Grape	Healthy	0.9956± 0.00110	0.9864± 0.00110	0.9642± 0.00101	0.973± 0.00102	0.9945
	Disease	0.9672± 0.00102	0.9763± 0.0014	0.9783± 0.00104	0.9883± 0.00103	0.9952
Squash	Healthy	0.9856± 0.00103	0.9753± 0.00103	0.974± 0.00101	0.963± 0.00103	0.9945
	Disease	0.9872± 0.00110	0.9752± 0.00119	0.9642± 0.00116	0.963± 0.00244	0.9933

Table 7 Average Performance of the Proposed NGSA-II Optimized EfficientNetV2 Nets in Edge Workstation(Raspberry Pi Model 4, and JETSON NANO Board)

Plant	Type	Performance Metrics				
		Accuracy	Precision	Recall	F1-score	AUC

Tomato	Healthy	0.98228± 0.00141	0.9563± 0.0012	0.9489± 0.0020	0.939± 0.00110	0.982
	Disease	0.978± 0.00110	0.9623± 0.00112	0.956± 0.0012	0.9503± 0.0012	0.980
Apple	Healthy	0.9773± 0.00104	0.963± 0.00142	0.960± 0.00101	0.9593± 0.00110	0.981
	Disease	0.9783± 0.00153	0.98± 0.00151	0.9702± 0.00102	0.972± 0.00103	0.980
Blueberry	Healthy	0.9835± 0.00120	0.973± 0.00109	0.964± 0.00110	0.9602± 0.0012	0.991
	Disease	0.9860± 0.00102	0.982± 0.00110	0.9764± 0.00102	0.966± 0.0012	0.992
Orange	Healthy	0.983± 0.00112	0.9733± 0.0012	0.970± 0.0011	0.9682± 0.00102	0.990
	Disease	0.980± 0.001152	0.9806± 0.00112	0.962± 0.0016	0.972± 0.00132	0.989
Pepperbell	Healthy	0.9810± 0.00110	0.9653± 0.00112	0.960± 0.00110	0.9602± 0.00112	0.988
	Disease	0.9809± 0.00112	0.9702± 0.00112	0.970± 0.00119	0.9632± 0.00012	0.973
Potato	Healthy	0.9755± 0.00102	0.9668± 0.00110	0.9545± 0.00112	0.9562± 0.00101	0.9830
	Disease	0.9652± 0.00102	0.9643± 0.00110	0.9520± 0.00102	0.9642± 0.00120	0.9830
Corn	Healthy	0.9729± 0.0012	0.9654± 0.00112	0.9640± 0.00102	0.96± 0.00102	0.9880
	Disease	0.9745± 0.00111	0.9505± 0.00106	0.9521± 0.00104	0.9753± 0.00110	0.9845
Raspberry	Healthy	0.9723± 0.00154	0.9602± 0.00121	0.9552± 0.00110	0.967± 0.00118	0.987
	Disease	0.9702± 0.00101	0.9620± 0.00101	0.9502± 0.0011	0.9653± 0.00102	0.988
Soyabeans	Healthy	0.9782± 0.00114	0.9742± 0.00103	0.970± 0.0012	0.963± 0.0010	0.988
	Disease	0.9856± 0.00103	0.9759± 0.00112	0.969± 0.00102	0.9653± 0.00102	1.00
Grape	Healthy	0.9934± 0.00118	0.9753± 0.0010	0.963± 0.00101	0.970± 0.00102	0.99
	Disease	0.9762± 0.00104	0.9652± 0.0012	0.9762± 0.00102	0.9553± 0.00102	0.99
Squash	Healthy	0.9882± 0.00102	0.9672± 0.00112	0.972± 0.00104	0.961± 0.00102	0.99
	Disease	0.9871± 0.00112	0.9667± 0.00108	0.9643± 0.00116	0.960± 0.00144	0.99

Table 6-7 clearly demonstrates the efficacy of the proposed surrogate classifier in classifying the different plant diseases. The NSGA-II optimized EfficientNetsV2 model has shown the better performance suitable for the different edge architectures.

4.2 Hardware Deployment Analysis

Figure7 demonstrates the deployment efficiency of the proposed crop disease detection model across CPU, Raspberry Pi, and Jetson Nano platforms. As shown in Figure (a), the inference latency decreases from 19.90 ms on the CPU to 15.55 ms on the Raspberry Pi and 12.90 ms on the Jetson Nano, demonstrating faster prediction execution on the edge GPU platform. Figure (b) presents the corresponding computation time, where the Jetson Nano achieves the lowest value of 112.89 ms compared with 127.90 ms for the CPU and 152.90 ms for the Raspberry Pi. The reported computational-load measurements in Figure (c) similarly indicate reduced computational demand for the Jetson Nano. Overall, the results demonstrate the proposed lightweight architecture's practical deployment capabilities across resource-constrained computing platforms, with the Jetson Nano exhibiting the most favorable execution characteristics among the evaluated devices.

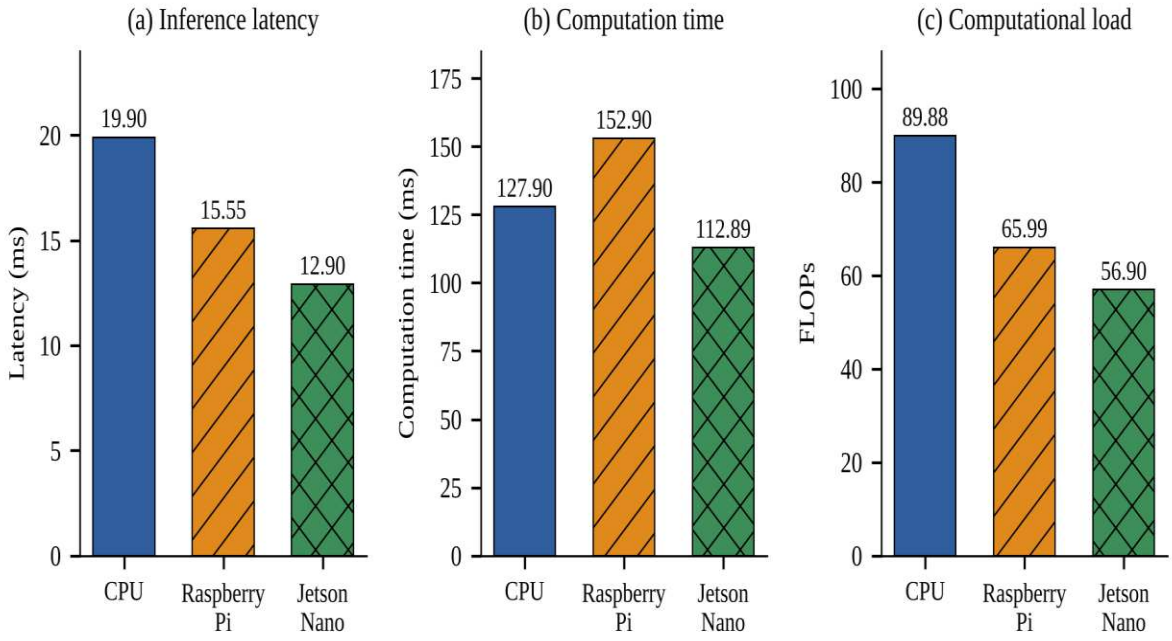


Figure 7 Deployment Analysis of the Proposed Framework a) Inference latency b) Computation time c) Computational Load.

4.3 Ablation Study Outcomes

Table 8 presents an ablation study used to evaluate the individual contributions of the proposed architecture-search and hardware-awareness components. The fixed EfficientNetV2 configuration is first considered as the baseline, followed by successive integration of NAS, multi-objective NSGA-II optimization, The baseline establishes the performance of the manually configured network, while the NAS configuration demonstrates the benefit of automatically optimizing architectural parameters. Incorporating multi-objective optimization further improves the accuracy–efficiency trade-off by simultaneously considering predictive performance and deployment constraints.The complete framework is therefore expected to provide the most favorable balance between disease-classification performance, computational complexity, inference latency, and memory consumption.

Table 8 Ablation study of the proposed architecture-search and hardware-awareness components

Configuration	NAS	NSGA-II	HW-aware	Accuracy (%)	Params (M)	FLOPs (M)	Latency (ms)
Baseline: fixed EfficientNetV2	✗	✗	✗	0.90	289393	103.89	22.89
+ NAS (accuracy only)	✓	✗	✗	0.91	200782	102.99	20.90
+ NAS + multi-objective NSGA-II	✓	✓	✗	0.88	189202	89.99	18.99
Proposed: full framework	✓	✓	✓	**0.982**	**175602**	**56.90**	**12.90**

4.4 Comparative Analysis

To prove the efficacy of the proposed model, baseline architectures and other existing lightweight models are considered for the experimentation. This study was conducted on Raspberry Pi Model 4B. Table 9 presents the performance of the different models in detecting the plant diseases. From the table 9, it is evident that the proposed model has shown the significant performance when compared with the other learning architectures.

Table 9. Ablation study of the proposed architecture-search and hardware-awareness components

Configuration	Accuracy (%)	F1-Score	AUC	Params (M)	FLOPs (M)	Latency (ms)
EfficientNetV2	0.90	0.852	0.88	289393	103.89	22.89
EfficientNetsB0	0.82	0.812	0.872	302922	120.32	27.73
MobileNetsV2	0.90	0.82	0.863	220682	102.99	28.88
ResNets-50	0.87	0.789	0.77	289204	172.99	32.99
Proposed Framework	**0.982**	**0.97**	**0.99**	**175602**	**56.90**	**12.90**

5 Conclusion and Future Enhancement

The research proposes the hardware-aware neural architecture search framework based on EfficientNetV2 for real-time crop disease detection on resource-constrained edge devices. The proposed framework integrates an EfficientNetV2-based searchable architecture space with NSGA-II to automatically optimize architectural parameters, including block type, kernel size, expansion ratio, network depth, input resolution, and width multiplier. The proposed evolutionary algorithms reduces exhaustive model training and hardware evaluation by enabling the efficient exploration of larger space. Experimental evaluation is carried out using PlantVillage Datasets and performance is measured across conventional and low-power

computing platforms. Experimental results demonstrates the proposed framework has achieved the accuracy of 0.982, precision of 0.97, recall of 0.96 , F1-score of 0.97 with latency of 12.90ms on Raspberry Pi Model. As the future development, the proposed framework needs improvisation in terms of deploying the dynamic and adaptive intelligence for diverse agricultural environments. Further evaluation using large-scale field datasets containing variations in illumination, occlusion, weather conditions, crop varieties, and disease severity will improve the robustness and generalization capability of the framework. In addition, online or continual learning can be incorporated to adapt the model to newly emerging diseases and changing environmental conditions without requiring complete retraining. Also the implementation of federated and privacy-preserving learning mechanisms could be explored to support collaborative model improvement across geographically distributed agricultural sites while retaining local farm data.

Data Availability Statement: Yes

Consent to Participate: Not applicable.

Research funding: No, this research did not receive funding.

Consent to Publish declaration missing: Not applicable.

Ethics declaration: Not applicable.

Author Contributions Statement

K. Anitha – Conceptualization, Data Collection
R. Naveenkumar – Methodology
Pooja H. L. – Software Development
Shanta kumar M – Validation and Evaluation
K. Valarmathi – Writing and Original Draft Preparation
K. M. Monica – Writing and Review and Editing

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

References

1] H. Slimani, J. El Mhamdi, and A. Jilbab, "Advanced algorithmic model for real-time multi-level crop disease detection using Neural Architecture Search," *E3S Web Conf.*, vol. 601, p. 00032, Jan. 2025, doi: 10.1051/e3sconf/202560100032.

[2] A. Y. Krishna, S. T. Sri, N. D. G, V. Sravya, P. S. Praneetha, and B. V. Vardhan, "Disease recognition of crops using ResNet and MDFC-ResNet," in *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, Erode, India, 2023, pp. 1109–1114, doi: 10.1109/ICSCDS56580.2023.10104714.

[3] S. Verma, P. Kumar, and J. P. Singh, "MLNAS: Meta-learning based neural architecture search for automated generation of deep neural networks for plant disease detection tasks," *Netw. Comput. Neural Syst.*, pp. 1582–1605, Jul. 2024, doi: 10.1080/0954898X.2024.2374852.

[4] R. Singh, L. K. Singh, A. K. Sharma, and M. Khanna, "Hybrid metaheuristic optimization of convolutional neural networks for tomato leaf disease classification," *Sci. Rep.*, vol. 16, Art. no. 25935, Jun. 2026, doi: 10.1038/s41598-026-25935-x.

[5] A. Umamageswari, N. Bharathiraja, and D. S. Irene, "A novel Fuzzy C-Means based Chameleon Swarm Algorithm for segmentation and Progressive Neural Architecture Search for plant disease classification," *ICT Express*, vol. 9, no. 2, pp. 264–270, Apr. 2023, doi: 10.1016/j.icte.2021.08.019.

[6] A. Sandhi, R. Kumar, R. Bhardwaj, D. Kumar, A. K. Rana, O. Ajala, A. Deepak, and A. O. Salau, "Optimized deep learning framework for pomegranate disease detection using nature-inspired algorithms," *Plant Methods*, vol. 21, Art. no. 124, Oct. 2025, doi: 10.1186/s13007-025-01240-x.

[7] İ. Yağ and A. Altan, "Artificial intelligence-based robust hybrid algorithm design and implementation for real-time detection of plant diseases in agricultural environments," *Biology*, vol. 11, no. 12, Art. no. 1732, Nov. 2022, doi: 10.3390/biology11121732.

[8] K. Kirti, N. Rajpal, J. Yadav, and K. K. Mondal, "Multi-resolution analysis and deep neural network architecture based hybrid feature extraction technique for plant disease identification and severity estimation," *Evol. Intell.*, vol. 17, no. 2, pp. 1163–1183, Apr. 2024, doi: 10.1007/s12065-022-00812-7.

[9] Y. Ye, Y. Chen, and S. Xiong, "Field detection of pests based on adaptive feature fusion and evolutionary neural architecture search," *Comput. Electron. Agric.*, vol. 221, Art. no. 108936, Jun. 2024, doi: 10.1016/j.compag.2024.108936.

[10] O. J. Omole, R. L. Rosa, M. Saadi, and D. Z. Rodriguez, "AgriNAS: Neural Architecture Search with Adaptive Convolution and Spatial–Time Augmentation Method for Soybean Diseases," *AI*, vol. 5, no. 4, pp. 2945–2966, Dec. 2024, doi: 10.3390/ai5040142.

[11] M. A. Saleem, M. Aamir, R. Ibrahim, N. Senan, and T. Alyas, "An optimized convolution neural network architecture for paddy disease classification," *Comput. Mater. Continua*, vol. 71, no. 3, pp. 6053–6067, 2022, doi: 10.32604/cmc.2022.022215.

[12] A. Birajdar, V. Dhuttarge, S. Degaonkar, A. Gurav, S. Garad, O. Bhave, A. Chatake, and N. Alange, "Efficient Convolutional Neural Network for Real-Time Crop Disease Detection in Precision Agriculture," *Int. J. Comput. Appl.*, vol. 187, no. 98, pp. 1–5, 2026, doi: 10.5120/ijca02c2de6188d7.

[13] A. Kalaivani, V. Asha, J. Sathya, R. Kumar, N. T. Ranganath, and R. Jain, "Smart agriculture: Federated and centralized AI models for real-time crop disease diagnosis," in *2025 6th International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2025, pp. 1–6, doi: 10.1109/ICIRCA65293.2025.11089601.

[14] H. Slimani, J. El Mhamdi, and A. Jilbab, "Deep learning structure for real-time crop monitoring based on Neural Architecture Search and UAV," *Braz. Arch. Biol. Technol.*, vol. 67, Art. no. e24231141, Sep. 2024, doi: 10.1590/1678-4324-2024231141.

[15] A. Uchitkar, B. Kolse, P. Ubhad, S. L. Nalbalwar, G. Kale, and N. Khandare, "Plant disease detection system using Convolutional Neural Networks: A comprehensive approach with real-time deployment," in *2025 9th International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, Pune, India, 2025, pp. 1–6, doi: 10.1109/ICCUBEA65967.2025.11283826.

[16] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Comput. Electron. Agric.*, vol. 153, pp. 46–53, Oct. 2018, doi: 10.1016/j.compag.2018.08.013.

[17] H. Slimani, J. El Mhamdi, and A. Jilbab, "Optimizing multi-level crop disease identification using advanced Neural Architecture Search in deep transfer learning," *Int. J. Comput. Digit. Syst.*, vol. 16, no. 1, pp. 1195–1207, Feb. 2024, doi: 10.12785/ijcds/160195.

[18] M. F. R. Asif, A. B. M. Sultan, M. S. Hossain, and M. A. Alam, "An optimized capsule neural network for tomato leaf disease detection and classification," *J. Vis. Commun. Image Represent.*, vol. 94, Art. no. 102620, Feb. 2023.