Journal of Agricultural Sciences
(Tarim Bilimleri Dergisi)

J Agr Sci-Tarim Bili
e-ISSN: 2148-9297
jas.ankara.edu.tr

A Deep Learning Model for Detection and Classification of Nutritional Deficiency in Coffee Plant

Umarani Chellamani[a]*, Baskaran Kaliaperumal[b]

[a]*Department of Computer Science and Engineering Alagappa Chettiar Government College of Engineering and Technology Karaikudi – 630 003, Sivagangai District, Tamilnadu, INDIA*

[b]*Department of Electrical and Electronics Engineering Alagappa Chettiar Government College of Engineering and Technology Karaikudi – 630 003, Sivagangai District, Tamilnadu, INDIA*

ARTICLE INFO
Research Article
Corresponding Author: Umarani Chellamani, E-mail: cumarani@accet.ac.in
Received: 17 October 2024 / Revised: 17 March 2025 / Accepted: 14 May 2025 / Online: 30 September 2025

Cite this article
Umarani C, Baskaran K (2025). A Deep Learning Model for Detection and Classification of Nutritional Deficiency in Coffee Plant. *Journal of Agricultural Sciences (Tarim Bilimleri Dergisi),* 31(4):960-980. DOI: 10.15832/ankutbd.1568929

ABSTRACT
Coffee is one of the most popular beverages consumed worldwide and is also an important economic driver in agricultural economies. However, nutritional deficiencies in coffee plants have a major effect on the quality and yield of the crop. Detection of these deficiencies early and accurately is critical for effective intervention and management. In this work, we introduce a novel deep-learning framework for detection and classification of nutritional deficiencies in coffee plants. DenseNet-201, AlexNet, and MobileNet-V2 are integrated to extract discriminative features from coffee leaf images, and an attention-based feature fusion mechanism is proposed using squeeze-and-excitation blocks to improve feature representation. A differential evolution algorithm is used to optimize a Kernel extreme learning machine for learning efficiency and generalization to classify the extracted features. A benchmark dataset is used for the evaluation of the proposed model and its performance is assessed against multiple performance metrics such as accuracy, precision, recall, specificity, F1-score, and the Matthews correlation coefficient. The proposed method is compared with existing deep learning models, and it is found that the proposed method outperforms the other models with a classification accuracy of 99.50%, precision of 99.22%, recall of 99.24%, specificity of 0.9947, F1-score of 0.9957, and M correlation coefficient of 0.9908. The model is able to identify nutritional deficiencies accurately, and these results confirm the model's effectiveness as a practical and scalable solution for precision agriculture and sustainable coffee cultivation.
.

Keywords: Coffee plant, Nutritional deficiency detection, Deep learning, Differential evolution algorithm, Kernel extreme learning machine, Classification

1. Introduction

The coffee trade is one of the most economically significant trades in the world, with millions of farmers depending on coffee cultivation for their livelihoods. Nevertheless, coffee plants' productivity and quality are highly affected by several factors, such as soil fertility, climatic conditions, and plant nutrition (Tsigkou et al. 2025). Physiological disorders, low yield, and poor quality of the bean are caused by nutritional deficiencies in coffee plants, which also affect the global coffee industry (Pillaca-Pullo et al. 2025). Timely corrective measures and crop loss minimization require early detection and accurate classification of these deficiencies for optimal agricultural management practices.

Current methods to detect nutritional deficiencies in coffee plants are manual and subjective, performed by agricultural experts, and are often time-consuming and prone to human error. Laboratory-based chemical analysis is precise, but expensive, laborious, and unfeasible for large-scale monitoring (Apraez et al. 2025). In recent years, artificial intelligence and deep learning have shown great potential in automating plant health diagnosis from leaf image analysis. Convolutional Neural Networks (CNNs) have proven to be very powerful tools for feature extraction and classification and as such can be used to accurately detect plant diseases and deficiencies with little or no human intervention.

Deep learning models have been extensively used for plant disease detection. However, each individual CNN model typically does not model the entire spectrum of complex visual patterns related to different types of nutritional deficiencies. Additionally, simple feature concatenation in the multi-model approaches may result in redundant or irrelevant information and subsequently deteriorate the classification performance (Upadhyay et al. 2025). In order to overcome these limitations, this work presents an enhanced deep learning framework that incorporates multiple feature extraction models with an attention-based feature fusion mechanism for improving classification accuracy.

1.1. Related works

Global coffee production is severely affected by coffee leaf diseases, which demand accurate detection and classification of the diseases. Deep learning has been the advent of disease identification in coffee plants by enabling the development of efficient and accurate classification models. This review summarizes the research works that use deep learning and machine learning methods for coffee disease detection.

CNN-based feature concatenation approach was used by Abuhayi & Mossa (2023) with GoogLeNet and ResNet, giving 99.08% accuracy of coffee disease classification. Preprocessing and augmentation of the images improved robustness. To achieve 84% accuracy with less computational complexity, Ramamurthy et al. (2023) developed a model based on an EfficientNetB0 with a ghost module. To achieve real-time classification, Yamashita & Leite (2023) embedded CNN models in microcontrollers resulting in 96% accuracy, making AI disease detection affordable for small-scale farmers. For hybrid feature fusion, Faisal et al. (2023) combined MobileNetV3, Swin Transformer, and VAE, which resulted in 84.29% accuracy and proved the efficiency of multi-model feature extraction. Coffee wilt disease detection, achieving 98.1% training and 97.9% test accuracy, was developed by Milke et al. (2023) using a CNN and highlighting the importance of dataset balancing and hyperparameter tuning.

Raghavendra (2023) proposed a colour-based filtering method for classification and severity prediction with machine learning techniques and feature extraction with 98.65% accuracy. In another research, Karthik et al. (2023) used an inception-based attention network for classification of the diseases with 98.57% accuracy and improved feature extraction by making use of global context as well as multi-head attention. Hasan et al. (2023) achieved 98% accuracy by using an automated clustering method based on kernel density estimation and ResNet50, which reduced the manual labelling errors. In Pham et al. (2023), the authors compared traditional AI methods to deep learning and concluded that ensemble CNN models are superior to single models for disease classification. In their work, Hitimana et al. (2023), trained multiple deep-learning models on a Rwandan dataset where DenseNet had the highest accuracy 99.57% and the result stresses the importance of dataset diversity. To classify the disease robustly, Nawaz et al. (2024) developed CoffeeNet based on an enhanced CenterNet approach with 98.54% accuracy and the ability to handle image distortions. A hybrid deep CNN (InceptionV3 + DenseNet121) was proposed by Singh & Kumar (2024) that reached 99% accuracy and surpassed the accuracy of single deep learning models. A dual path visual transformer for real-time lesion segmentation and classification was proposed by Salamai (2024) to increase the diagnosis accuracy.

SUNet (Thakur et al. 2024) combines U-Net and SegNet for segmentation and classification, which outperforms existing approaches on various metrics. Coffee-Net, a deep mobile patch generation network with 99.95% accuracy using Region-based Convolutional Neural Networks (RCNN) based patch segmentation was proposed by Bidarakundi & Kumar (2024) and de Oliveira Aparecido et al. (2024) used machine learning (XGBoost) to predict Phoma leaf spot outbreak using climate data to predict early disease. Shrma & Rahiman (2024) compared CNN and SVM in disease prediction, CNN yielded an accuracy and effectiveness of 95.8% whereas SVM was 79.6%. In Demilie (2024), plant disease classification techniques were analysed and CNNs were proven to be the most effective because they can learn hierarchical features. CoffNet developed by Adelaja & Pranggono (2025) is able to detect diseases in real time with 99.89% accuracy and classify 125.93 frames per second. In Karia et al. (2025), DenseNet201 was tested on other datasets and high-quality images were found to improve coffee leaf rust detection accuracy. de Souza Dias et al. (2025) proposed a novel segmentation and near-sets clustering to detect coffee diseases by using multi-scale segmentation. The above frameworks for coffee disease detection are summarized in Table 1.

Various machine learning and deep learning algorithms have been explored for plant leaf disease classification across different countries, environmental conditions, and plant species, such as pepper (Bhagat & Kumar 2023; Rahman et al. 2025), lemon (Saygılı 2023), rice (Haridasan et al. 2023; Shovon et al. 2023; Sankareshwaran et al. 2023; Preethi et al. 2024), tomato (Chug et al. 2023; Barman et al. 2024; Wang & Liu 2025), grape (Sofuoğlu & Bırant 2023), potato (Sofuoğlu & Bırant 2024), corn (Idress et al. 2024), cucumber (Ali et al. 2024), soybean (Duhan et al. 2025), wheat (Baiju et al. 2025), barley (Rezaei et al. 2025), cassava (Dosset et al. 2025).

Table 1- Summary of frameworks for coffee disease detection

Authors (Year)	*Method used*	*Dataset used*	*Metrics used*	*Contribution*
Abuhayi & Mossa (2023)	CNN with GoogLeNet & ResNet	Custom dataset	Accuracy	High-accuracy CNN model with feature concatenation
Ramamurthy et al. (2023)	EfficientNetB0 with ghost module	Arabica coffee dataset	Accuracy	Low-complexity EfficientNet model for coffee disease classification
Yamashita & Leite (2023)	CNN on microcontrollers	Custom dataset	Accuracy	Affordable real-time disease detection on microcontrollers
Faisal et al. (2023)	Hybrid (MobileNetV3, Swin Transformer, VAE)	RoCoLe dataset	Accuracy	Hybrid feature fusion for improved classification
Milke et al. (2023)	CNN for wilt disease detection	4000 coffee leaf images	Accuracy	High-accuracy wilt disease detection model
Raghavendra (2023)	Color-based filtering and ML	Custom dataset	Accuracy	Novel color-based filtering for severity prediction
Karthik et al. (2023)	Inception-based attention network	BRACOL dataset	Accuracy, F1-score	Attention-based deep learning approach for disease classification
Hasan et al. (2023)	Kernel density clustering with ResNet50	Coffee leaf datasets	Accuracy	Automated clustering to reduce annotation errors
Pham et al. (2023)	Ensemble CNN models	Multiple datasets	Accuracy	Ensemble learning for higher classification accuracy
Hitimana et al. (2023)	DenseNet on Rwandan dataset	Rwandan dataset (37,939 images)	Accuracy	DenseNet-based best-performing model
Nawaz et al. (2024)	CoffeeNet (CenterNet + ResNet50)	Arabica coffee leaf dataset	Accuracy, mAP	Enhanced CenterNet-based CoffeeNet model
Singh & Kumar (2024)	Hybrid CNN (InceptionV3 + DenseNet121)	Plant Village dataset	Accuracy	Hybrid CNN for superior classification
Salamai (2024)	Dual-Path Visual Transformer (COFFORMER)	Custom dataset	Improved classification precision	Dual-path visual transformer for segmentation and classification
Thakur et al. (2024)	SUNet (U-Net + SegNet)	JMuBEN & JMuBEN2 datasets	IoU, F1-score, precision, recall	Hybrid deep learning segmentation model
Bidarakundi & Kumar (2024)	Coffee-Net with RCNN patch segmentation	Multiple datasets	Accuracy	Patch-based RCNN segmentation approach
de Oliveira Aparecido et al. (2024)	XGBoost with climate data	Brazilian coffee farms	R^2, RMSE	Machine learning-based early disease prediction using climate data
Shrma & Rahiman (2024)	CNN vs SVM for disease prediction	Synthetic datasets	Accuracy	Comparison of CNN and SVM for plant disease detection
Demilie (2024)	Comparison of ML & DL techniques	Comparative analysis datasets	Performance comparisons	Evaluation of various ML and DL approaches
Adelaja & Pranggono (2025)	CoffNet for real-time detection	Mixed coffee leaf datasets	Accuracy, Processing Speed	Real-time, high-speed coffee disease detection model
Karia et al. (2025)	DenseNet201 with dataset quality analysis	Mbozi & custom datasets	Accuracy	Impact of dataset quality on classification performance
de Souza Dias et al. (2025)	JSEG segmentation & near sets clustering	RoCoLe dataset	Segmentation accuracy	Improved segmentation-based classification using JSEG and clustering

The novelty of this study is summarized as follows:

1. This study differs from traditional single CNN-based models by combining DenseNet-201, AlexNet, and MobileNet-V2 to extract multi-scale discriminative features. The novel combination of these architectures exploits the complementary strengths of each architecture in order to better recognize subtle leaf texture and color variations related to nutritional deficiencies.
2. A novel feature fusion strategy with squeeze-and-excitation blocks is introduced instead of conventional methods of feature concatenation often leading to redundant information. This mechanism dynamically recalibrates the feature importance such that the model pays more attention to the most important features, and less to the irrelevant ones, leading to dramatic acceleration in the classification accuracy.
3. The proposed classifier method is different from conventional deep learning classifiers that depend on backpropagation-based training. The proposed classifier uses a Kernel extreme learning machine that does not require iterative weight tuning and is a single-pass learning method. Such novel integration can significantly reduce training time without compromising classification performance.

4. To enhance the classification accuracy, a new hyperparameter optimization strategy based on a differential evolution algorithm is introduced to the Kernel extreme learning machine. In contrast to traditional optimization methods, the differential evolution algorithm provides better global search capabilities, facilitating a better generalization and robustness of the models in finding different deficiency classes.

The objective of this study is to develop an advanced deep-learning framework for the detection and classification of nutritional deficiencies in coffee plants.

2. Material and Methods

This paper introduces a multi-model deep learning model for coffee plant nutritional deficiency detection and classification using attention-based feature fusion and an optimized classification mechanism. As shown in Figure 1, the architecture includes several key components: preprocessing, feature extraction, attention-based feature fusion, Kernel Extreme Learning Machine (ELM) for classification, and hyperparameter optimization by differential evolution algorithm.

Image preprocessing is the first step of the process; raw coffee leaf images are subjected to weighted averaging operation-based guided filtering and contrast enhancement to enhance the clarity and important features of the images. This step makes sure that the input data is very well optimized for feature extraction by removing noise and improving relevant structures. After preprocessing, features are extracted via a multi-model approach using three pre-trained deep learning architectures namely DenseNet-201, AlexNet, and MobileNet-V2. These architectures contribute unique strengths in each of these models. DenseNet-201 is capable of extracting deep hierarchical features. This guarantees that the model learns fine-grained as well as high-level patterns related to various nutritional deficiencies. AlexNet is good at capturing basic structural features, and MobileNet-V2 is lightweight but powerful.

An attention-based feature fusion mechanism using squeeze-and-excitation blocks is applied to improve the robustness of the feature representation. Consequently, squeeze-and-excitation blocks recalibrate the importance of extracted features dynamically, assigning adaptive weights to every feature map, such that the weights are adaptively decided to emphasize the important features and suppress the redundant information instead of simple feature concatenation. The attention-driven fusion strategy enables the model to pay attention to the most discriminative parts of the deficiency patterns and hence improve the classification accuracy.

The Kernel ELM classifier is then selected as it has high-speed learning capability and better generalization performance than conventional backpropagation-based classifiers, and the fused features are fed into them. In contrast to the standard neural networks that need to adjust weights iteratively, Kernel ELM computes output weights analytically in a single-pass learning process which is computationally efficient. To improve the classification performance, a differential evolution optimization algorithm is utilized to optimize the hyperparameters of Kernel ELM. The model is enhanced by using a powerful evolutionary algorithm, i.e. differential evolution, which helps in efficiently choosing the best hyperparameters so as to make the classifier achieve an optimal tradeoff between accuracy and computation efficiency.

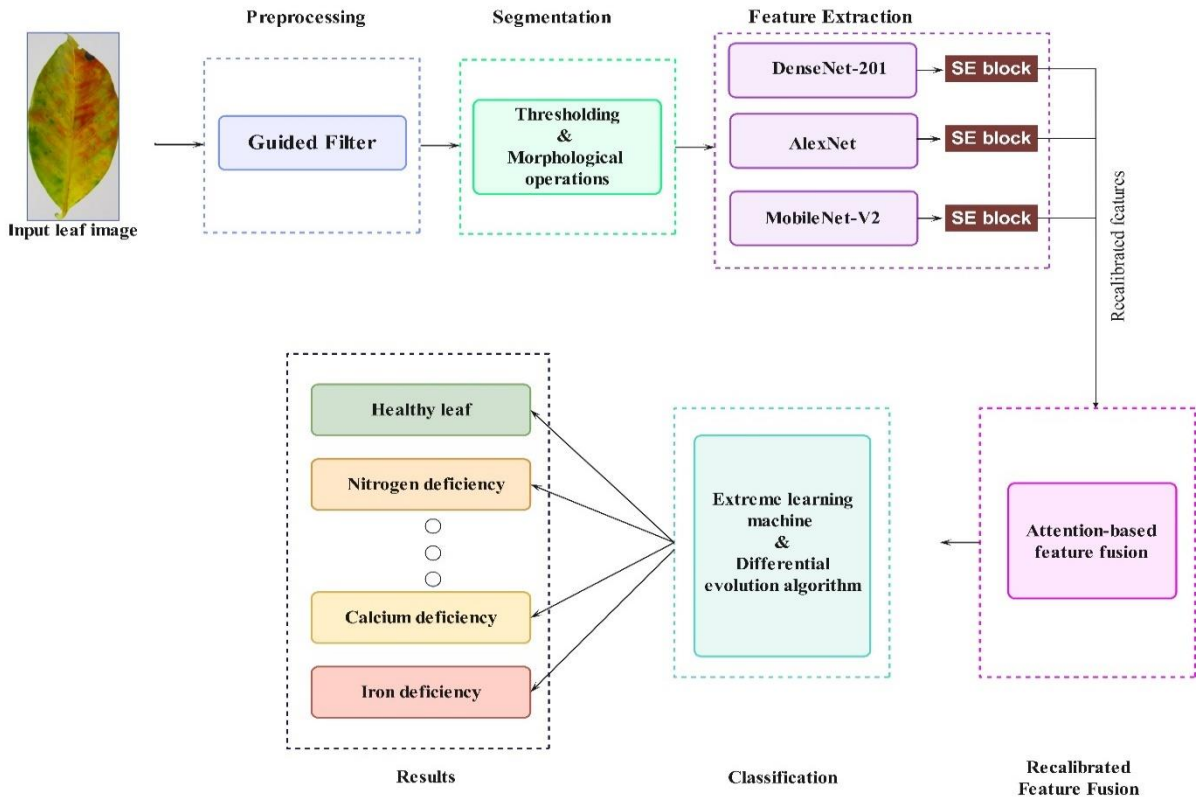


Figure 1- Proposed model for detection and classification of nutritional deficiency in coffee plant

In the following sub-sections, the features of each component of the proposed model are described in detail in terms of how it improves feature extraction, fusion, and classification to predict nutritional deficiency in coffee plants.

2.1. Weighted averaging operation-based guided filter process

It is an image preprocessing technique consisting in removing the noise from the image and preserving edges and structural details. It is especially effective for denoising, edge-preserving smoothing, and tone mapping. The concept of the guided filter is based on the idea that a guidance image (which is usually a smoothed version of the input image) is used to regulate the filtering process. This keeps large structures while minimizing unwanted distortions and noise.

Two inputs are fed into the filtering process (Sandhiya & Raja 2024), one is the original input image that needs to be processed and the other is the guidance image that supplies the structural information for filtering. The main idea of the guided filter mechanism is local filtering, in which a filtering window is defined around each pixel in the input image. The filter calculates linear coefficients within this window that dictate the effect of the neighboring pixels on the output value of the target pixel. The coefficients of these filters are computed according to the intensity values of the input and guidance images, such that the filter localizes to image structures and does not apply uniform smoothing.

The guided filter coefficient at each pixel location is computed using the following equation:

$$a_{ij} = \frac{1}{n}\sum_{p\in\Omega}\left(\frac{I_{ip}-\bar{I}_i}{\sigma_i^2+\epsilon}\right)\left(\frac{P_{ip}-\bar{p}_i}{\sigma_p^2+\epsilon}\right) \quad (1)$$

Where; I_{ip} is the intensity value of the input image at the pixel p within the filtering window Ω. P_{ip} is the corresponding intensity value of the guidance image at the pixel p within the filtering window Ω. $\bar{I}_i$ and $\bar{p}_i$ are the means of the intensity values of the input and guidance images, respectively, within the filtering window Ω. σ_i^2 and σ_p^2 are the variances of the intensity values of the input and guidance images, respectively, within the filtering window Ω. n is the number of pixels within the filtering window Ω. ϵ is a small positive value added to the variances to prevent division by zero.

The coefficient a_{ij} represents the degree of linear relationship between the input image and the guidance image within the filtering window Ω normalized by the variances of both images and adjusted by the small positive value ϵ to stabilize the computation.

The guided filter performs a weighted average operation using the computed coefficients and pixels that are more similar to the guidance image will receive higher weights. The following equation is used to perform the weighted averaging operation.

$$q_i = \bar{P}_i + \frac{1}{n}\sum_{p\in\Omega} a_{ij}\left(I_{ip} - \bar{I}_i\right) \quad (2)$$

Where; q_i is the filtered output value for the pixel i in the input image. $\bar{P}_i$ is the mean intensity value of the guidance image within the filtering window Ω centered at pixel i. I_{ip} is the intensity value of the input image at the pixel p within the filtering window Ω. $\bar{I}_i$ is the mean intensity value of the input image within the filtering window Ω centered at pixel i. a_{ij} is the coefficient computed for pixel i in the guided filter process, representing the degree of linear relationship between the input and guidance images within the filtering window Ω. n is the number of pixels within the filtering window Ω. This formula computes the weighted sum of the differences between the input image intensity values I_{ip} and their mean $\bar{I}_i$, weighted by the coefficients a_{ij}, and adds this sum to the mean intensity value $\bar{P}_i$ of the guidance image within the window.

This operation effectively removes the noise from the image and keeps the important structures, namely leaf veins, textures, and contours, which are required for accurate classification. Since the process uses local information from both the input and guidance images, it is highly adaptive to variations in intensity and texture and hence is ideally suited to preprocessing tasks in plant image analysis. A guided filtering process is usually an iterative process with multiple passes to further refine the filtering results. The resulting output is a version of the input image, which is noise-reduced and edge-preserved and improves the visibility of nutritional deficiency patterns in coffee plant leaves.

2.2. Thresholding and morphological operations

Image segmentation to separate the leaf from the background is performed by thresholding and morphological operations. A simple but powerful technique is thresholding for segmenting objects in an image based on pixel intensity values. The image is thresholded by selecting a threshold value and comparing each pixel in the image to the threshold. Based on the pixel value, above or below the threshold, it is classified as either foreground (object) or background. Global thresholding and local thresholding are the two types of thresholding. Global thresholding uses a single threshold value for all the pixels in the image. In local thresholding, different threshold values are used for different regions of the image based on local variations in illumination and contrast. This work uses adaptive thresholding.

Morphological operation is the operation of manipulation of the geometric structure in the image. It is a few basic operations. This operation first shrinks the boundaries of foreground objects by removing pixels from the edges. Secondly, it increases the foreground object boundaries by adding pixels at the edges. Thirdly, it is erosion, followed by dilation. It is good for removing small objects or noise without losing the shape or size of larger objects. Lastly, it does dilation and then erosion. It is useful for filling small gaps or holes within objects. In image segmentation morphological operation is very useful for cleaning up segmented images, removing noise, filling gaps, and separating connected objects. When the segmentation of an object from the background is done by applying thresholding, the morphological operation is used to refine the segmentation. The background portion of the object image is effectively removed using thresholding and morphological operations to give a cleaner and more accurate segmentation.

2.3. DenseNet-201

DenseNet-201 is an improved model by concatenating features to improve feature summation (Wang & Zhang 2020). Mathematically, this process can be expressed as follows:

$$\mathrm{I_m} = \mathrm{Z_m}(\mathrm{s_{m-1}}) \quad (3)$$

$$\mathrm{I_m} = \mathrm{Z_m}(\mathrm{s_{m-1}}) + \mathrm{s_{m-1}} \quad (4)$$

$$\mathrm{I_m} = \mathrm{Z_m}([\mathrm{s_0}, \mathrm{s_1}, \mathrm{s_2}, \ldots, \mathrm{s_{m-1}}]) \quad (5)$$

In the above equations, $\mathrm{I_m}$ represents the m^{th} layer feature, Z represents the nonlinear operation, and m represents the layer index.

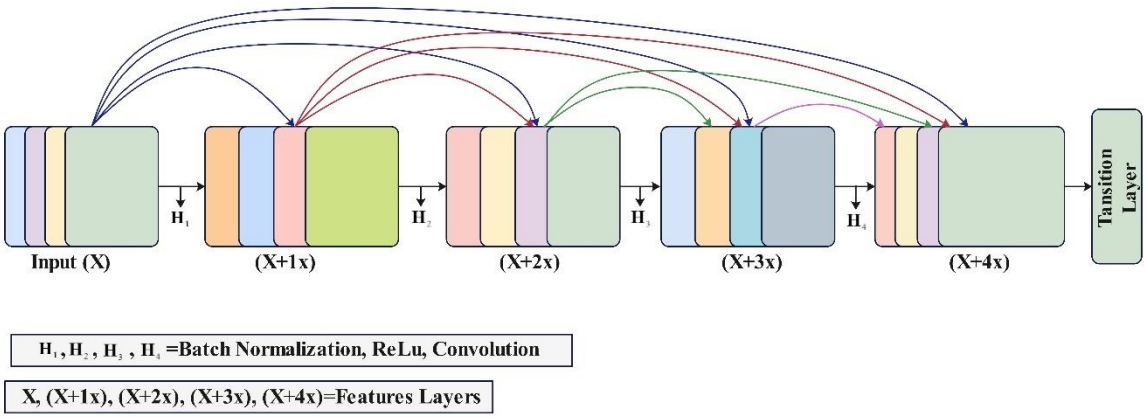


Figure 2- Architecture of DenseNet-201

This architecture enhances feature propagation, leading to more detailed and discriminative representations of complex patterns. In particular, DenseNet-201 is useful at capturing hierarchical and fine-grained details of coffee leaf images and hence it is suitable for the detection of subtle deficiencies like marginal chlorosis, interveinal yellowing, and necrotic spots. As shown in Figure 2, DenseNet-201 has an architectural layout with skip connections within each block to accelerate the speed of the classification process. Batch normalization, pooling, ReLU, convolutional layers, and lastly, classification by the SoftMax classifier form a block. Our proposed approach features a denseNet-201 as a feature extractor where features are chosen up to the final convolution layer and integrated with AlexNet and MobileNet-V2.

Table 2- Detailed information for the DenseNet201

Layer	***Details***	***Output***
Pooling	3 × 3 Max-pool, Stride2	56×56
Convolution	7 × 7 Conv, Stride2	112×112
DenseBlock1	$\begin{bmatrix}1x1\ Conv\\3x3\ Conv\end{bmatrix}$ x6	56×56
TransitionLayer1	1x1Conv 2 × 2 avg-pool, Stride2	56×56 28×28
DenseBlock2	$\begin{bmatrix}1x1\ Conv\\3x3\ Conv\end{bmatrix}$ x12	28×28
TransitionLayer2	1 × 1 Conv 2 × 2 avg-pool, Stride2	28×28 14×14
DenseBlock3	$\begin{bmatrix}1x1\ Conv\\3x3\ Conv\end{bmatrix}$ x48	14×14
TransitionLayer3	1 × 1 Conv 2 × 2 avg-pool, Stride2	14×14 7×7
DenseBlock4	$\begin{bmatrix}1x1\ Conv\\3x3\ Conv\end{bmatrix}$ x32	7×7
ClassificationLayer	7 × 7 global average pool 1000D fully connected, softmax	14×14

The detailed architectural configuration of the DenseNet-201 model is presented in Table 2 and contains layer-wise structure and key operations. DenseNet-201 is a deep convolutional neural network with dense connectivity where each layer receives inputs from all preceding layers to improve gradient flow and feature reuse. The network starts with an initial convolutional layer with a stride of 2 and a max-pooling layer that decreases spatial dimensions while maintaining significant features. The model is composed of four dense blocks containing multiple convolutional layers with dense connectivity. Transition Layers are used between these blocks which consist of 1×1 convolutions for feature reduction and average pooling layers for down-sampling. These transition layers serve to control computational complexity at the expense of some feature integrity. The last stage of the network is a global average pooling layer that averages spatial information to a single vector representation. The output of this is then fed into a fully connected layer with 1000 output neurons and a softmax activation function, which is suitable for multi-class classification problems. DenseNet-201 architecture is efficient in feature reuse and has a compact design, which makes it ideal for deep feature extraction in complex classification problems such as detecting nutritional deficiencies in coffee plants.

2.4. AlexNet

The seminal transfer learning model, AlexNet by Krizhevsky et al. (2017) is applied in a number of computer vision tasks. Although it is a simple structured model, it is a good feature extractor because of its deep hierarchical architecture that efficiently

extracts low and mid-level features like edges, textures, and color variations. Within the context of coffee leaf deficiency detection, AlexNet assists in characterizing basic structural patterns that are significant in discerning the leaf textures and discolored areas induced by nutritional deficiencies. As shown in Figure 3, this network architecture consists of 8 weighted layers including 5 convolutional layers followed by 3 fully connected layers. At the input layer, the network ingests images of dimension 224×224×3. The first convolutional layer has ninety-six 11 × 11 kernels with stride 4, and the next one has 256 5 × 5 kernels. Max pooling operation follows each convolutional layer. The next layers are a convolutional layer with 384 kernels of size 3 × 3, another with 384 kernels of the same size, and the fifth convolutional layer with 256 kernels, without intermediate pooling layers. A max-pooling layer and two fully connected layers with 4096 neurons each are then followed. The last layer of the network is a softmax layer with 1000 class labels. To speed up computation, the ReLU activation function is used throughout the network, and to prevent overfitting, dropout is used.

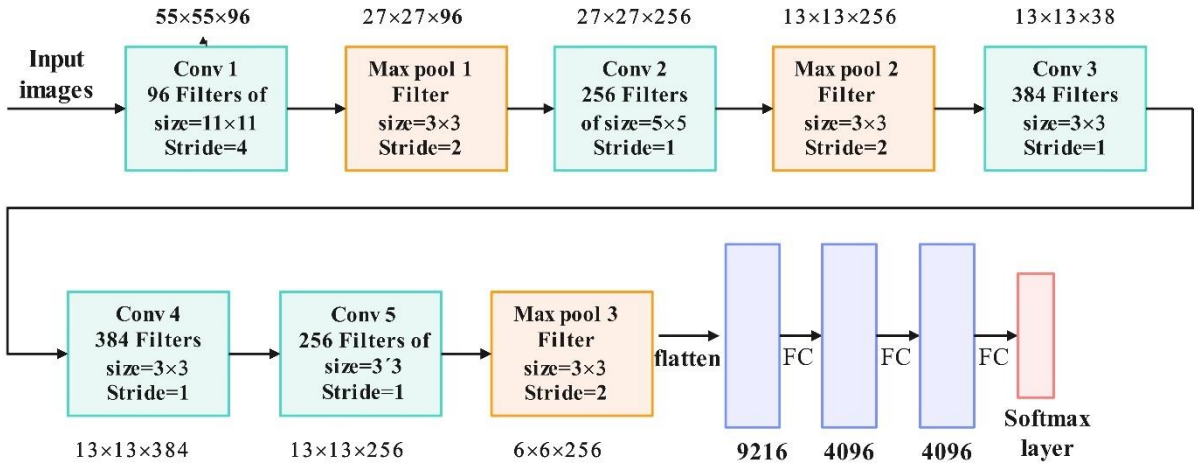


Figure 3- Architecture of AlexNet

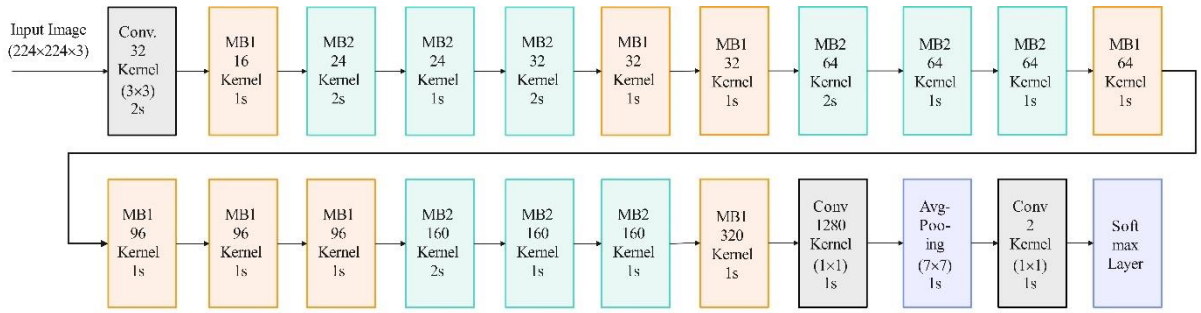


Figure 4- Architecture of MobileNet-V2

2.5. MobileNet-V2

MobileNet-V2 is a convolutional neural network architecture designed with utmost care for mobile and embedded vision applications. It is known for its efficiency and effectiveness in transfer learning and brings forward an ingenious inverted residual design with skip connections between thin bottleneck layers (Sandler et al. 2018). To make the proposed framework computationally efficient and possibly for real-time deployment in agricultural settings, this model is selected. DenseNet 201 and AlexNet capture deep feature representations and MobileNet-V2 captures high-level abstract feature representations with lower computational complexity. Each layer of Figure 4 consisting of three convolution layers and seventeen bottleneck layers contains detailed attributes that work in concert to efficiently extract features from input data. Moreover, MobileNet-V2 uses the ReLU6 activation function, which brings non-linearity, a crucial part of learning complex patterns, especially in cases when low-precision computation is required along with the need of robustness. MobileNet-V2 is a versatile and potent transfer learning option for applications, such as coffee leaf deficiency detection and classification, based on the amalgamation of architectural ingenuity and computational enhancements.

2.6 Feature extraction and attention-based feature fusion

Coffee leaf images are classified to identify a nutritional deficiency, and feature extraction is of paramount importance in identifying patterns. However, concatenating features as is may bring redundancy and miss the most important aspects of

correctly classifying. To address this limitation, an attention-based feature fusion mechanism is introduced with Squeeze-and-Excitation (SE) blocks to recalibrate the feature importance prior to fusion.

The proposed approach recalibrates each extracted feature set with the SE block. This starts with the squeeze operation, which aggregates the global spatial information through global average pooling to obtain a compact representation of the feature map. This operation guarantees the preservation of essential channel-wise dependencies to keep relevant structural information in the leaf images. The output of this step is then passed through an excitation operation which is a fully connected layer with sigmoid activation to learn channel-wise attention scores. These scores indicate the relative importance of the different feature channels and enable the network to magnify the important features and suppress the less informative ones. Finally, the scaled version of the original feature maps is obtained by multiplying the feature maps with the learned attention weights, producing recalibrated feature representations that emphasize the most discriminative patterns.

Given an input feature map '*X'* from any of the backbone models (DenseNet-201, AlexNet, MobileNet-V2), the SE attention weight vector '*s'* is computed using the squeeze and excitation steps as follows:

The SE block first compresses spatial information into a compact vector by applying global average pooling. The channel descriptor '*s*' is computed for each feature channel '*c*' as:

$$s_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_c(i,j) \tag{6}$$

Where; *H* and *W* represent the spatial dimensions of the feature map, and $X_c(i, j)$ denotes the activation value at position (*i, j*) for channel *c*. This operation aggregates global spatial information, enabling the model to retain the overall importance of each feature channel.

The squeeze operation is followed by a fully connected network that learns channel-wise attention scores using the pooled features. The first transformation is made of two fully connected layers with ReLU and sigmoid activations to make sure that the output is bounded and nonlinear:

$$e = \sigma\big(W_2 \cdot \delta(W_1 \cdot s)\big) \tag{7}$$

Where; W_1 and W_2 are trainable weight matrices, δ is the ReLU activation function, σ is the sigmoid activation function that normalizes the attention scores between 0 and 1. This step guarantees that the recalibration mechanism learns the relative importance of each channel according to its contribution towards the classification task. The learned attention weights e_c are then used to scale the original feature maps, adjusting their importance dynamically. The recalibrated feature map X_c' is computed as:

$$X_c' = e_c \cdot X_c \tag{8}$$

Where; X_c is the original feature map, and e_c is the corresponding learned attention score. This operation amplifies essential features while suppressing redundant or irrelevant ones, ensuring that the fused representation is more informative.

The SE block is applied to each feature set extracted from DenseNet-201, AlexNet, and MobileNet-V2, and the recalibrated features are concatenated to create a global representation. The Kernel ELM classifier is optimized using a differential evolution algorithm to ensure robust classification of nutritional deficiencies and this fused representation is passed into the Kernel ELM classifier. The SE blocks are integrated into the model to dynamically focus on the most relevant feature channels and improve the classification accuracy by removing the noise and redundant information. Finally, the proposed attention-based fusion strategy further enhances the overall effectiveness of the model, making sure that the classification is performed using the most important elements of the extracted features and not on the raw concatenation.

2.7. Classification using kernel ELM

In this study, the Kernel ELM is used as the classifier because of the reasons that it has a fast learning speed, improved generalization capability, and is efficient in dealing with high-dimensional feature spaces. Unlike the traditional deep learning classifiers, which depend upon iterative gradient-based learning, ELM is one-pass learning that randomly assigns the input weights and biases and the output weights are computed analytically. This leads to a much shorter training time with respect to backpropagation-based networks. (Sandhiya & Raja 2024). ELM comprises a specific hidden-layer feed-forward network, wherein the output is determined analytically according to randomly assigned input weight. The hidden neuron layer incorporates activation functions such as hard limiting, sine, Gaussian, and sigmoid, functions. The output neuron utilizes a linear activation function. Notably, ELM outperforms gradient-based learning models and traditional feed-forward neural networks in terms of avoidance of the local optimal solutions, generalization ability, and learning speed. Mathematically, the kernel ELM model encompasses N samples denoted as $\{P_i, Q_i\}$, where each sample i is represented by an n-dimensional feature $P_i = [p_{i1}, p_{i2}, .., p_{in},]$, and the class label is indicated as $Q_i = [q_{i1}, q_{i2}, .., q_{ic},]$. Assigning a class label l_k to sample P_i results in the k^{th} element of Q_i being set to one. A negative value (-1) is assigned to other elements.

The output of the ELM model with hidden neurons (H) and distinct classes (C) is formulated as follows.

$$\hat{q}_{ik} = \sum_{j=1}^{H} U_{kj} G_j\,(W, b, P_i) k = 1,2,..,C \quad (9)$$

In this equation, b represents the hidden neuron bias, W represents the input weight, u_{kj} represents the output weight function. The output of the hidden neuron, which is represented as $G_j(.)$, is written as given below.

$$G_j = G\left(\sum_{k=1}^{n} W_{jk} P_{ik} + b_j\right) j = 1,2,\dots,H \quad (10)$$

In the above equation, the activation function is represented as $G\ (.)$. For the radial basis function, the neuron output $G_j(.)$ is written as

$$G_j = G\left(b_j \|P - W\|\right) j = 1,2,\dots,H \quad (11)$$

In this equation, W represents the width of the radial basis function, b_j represents the neuron center of the radial basis function. The neuron output $G_j(.)$ for the sigmoid function is written as

$$G_j = tanh\left(\mathrm{b_j} + \sum_{\mathrm{k=1}}^{\mathrm{n}} \mathrm{W_{jk}\ p_{ik}}\right) \mathrm{j} = 1,2,\dots,\mathrm{H} \quad (12)$$

The hidden layer output Q_H can be included as follows.

$$\hat{Q} = UQ_H \quad (13)$$

The above equation can be represented in matrix form.

$$Q_H(W, B, P) = \begin{bmatrix} G_1(W, b_1, p_1) G_1(W, b_1, p_2) & \dots & G_1(W, b_1, p_n) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ G_H(W, b_H, p_1) G_H(W, b_H, p_2) & \dots & G_H(W, b_H, p_n) \end{bmatrix} \quad (14)$$

In this equation, $\hat{Q}$ represents the coded labels. Depending on the submerged neuron quantity, the bias of hidden neurons B and the key to weights W are arbitrarily selected.

The weight is analytically formulated as given below.

$$\hat{U} = QQ_H^{+} \quad (15)$$

Where; Q_H^{+} is the generalized pseudo inverse factor of the hidden layer output matrix.

The ELM algorithm is delineated as follows. Firstly, activation functions and hidden neurons are chosen for the classification problem. Next, input weights and biases are randomly selected, followed by the calculation of output weights. Subsequently, class labels are estimated utilizing the calculated bias weight, output, and hidden neuron. To optimize the parameters of the network, the differential evolution algorithm is integrated.

2.8. Differential evolution algorithm

The differential evolution algorithm is a global optimization technique that is very efficient in finding near-optimal solutions with minimal computational cost. Differential evolution algorithm differs from the traditional optimization methods since it works based on mechanisms like mutation, crossover, and selection, which enables it to explore the search space to a greater extent as well as prevents it from getting stuck in local optima (Wang et al. 2025). The differential evolution algorithm employs a population-based global search technique. This algorithm uses one-on-one competition and reduces the complexity of mutation operations.

By adjusting candidate solutions, it explores various solutions in parallel, dynamically tracking the current search using its memory capacity to adapt a search technique. This approach facilitates global convergence and robustness in the optimization process.

In mathematical terms, there are N candidates in the population of each generation G. The candidate can be written as $X_{iG} = (x_{iG}^1, x_{iG}^2, \dots, x_{iG}^M), i = 1, 2, \dots, N$, where M is the number of features.

The initial population is generated at random, spanning the search interval's lower and upper bounds.

$$X_i^n = X_i^{n,L} + rand(\ \ ) \cdot \left(X_i^{n,U} - X_i^{n,L}\right), = 1,2,\dots,M,\ n = 1,2,\dots,N, \tag{16}$$

Where; $X_i^{n,U}$ represents the upper bound of the variable X_i^n , and $X_i^{n,L}$ represents the lower bound of the variable X_i^n.

Three vectors $X_{r1,G}$, $X_{r2,G}$, $X_{r3,G}$ are selected to execute the mutation process. The following formula is implemented:

$$V_G^n = X_{r1,G}^n + F \cdot \left(X_{r2,G}^n - X_{r3,G}^n\right), \tag{17}$$

Where; V_{G+1}^n is the donor vector, $F \in [0,1]$ is the variation factor that regulates the amplification degree of the differential variable. $X_{r2,G}^n - X_{r3,G}^n$.

In the recombination process, the operator uses the following formula in order to develop the trial vector $U_{i,G+1}^n$ from the donor V_{G+1}^n and target $X_{i,G}^n$ vectors (Yazdani et al. 2024).

$$U_{i,G+1}^n = \begin{cases} V_{G+1,}^n if\ rand(\ \ ) \gg C_p\ or\ i = I_{rand} \\ X_{i,G}^n, if\ rand(\ \ ) > C_p\ and\ i \neq I_{rand} \end{cases}, \tag{18}$$

Where; $i = 1,2,\dots M, n = 1,2,\dots,N$, C_p represents the recombination probability, and $I_{rand} \in [1,M]$ represents a random number. Recombination is a mechanism that enables a portion of the code from the previous and new candidate solutions to swap places, creating a new individual.

The selection process begins following the completion of mutation and recombination processes.

The trial $U_{i,G+1}^n$ and target $X_{i,G}^n$ vectors are compared.

For the next generation, the vector with the lowest fitness value is chosen.

$$X_{i,G+1}^n = \begin{cases} U_{i,G+1}^n,\ if\ f\left(U_{i,G+1}^n\right) < f\left(X_{i,G}^n\right) \\ X_{i,G}^n,\ Otherwise \end{cases}, \tag{19}$$

Where; $i = 1,2,\dots,M$, and $n = 1,2,\dots,N$.

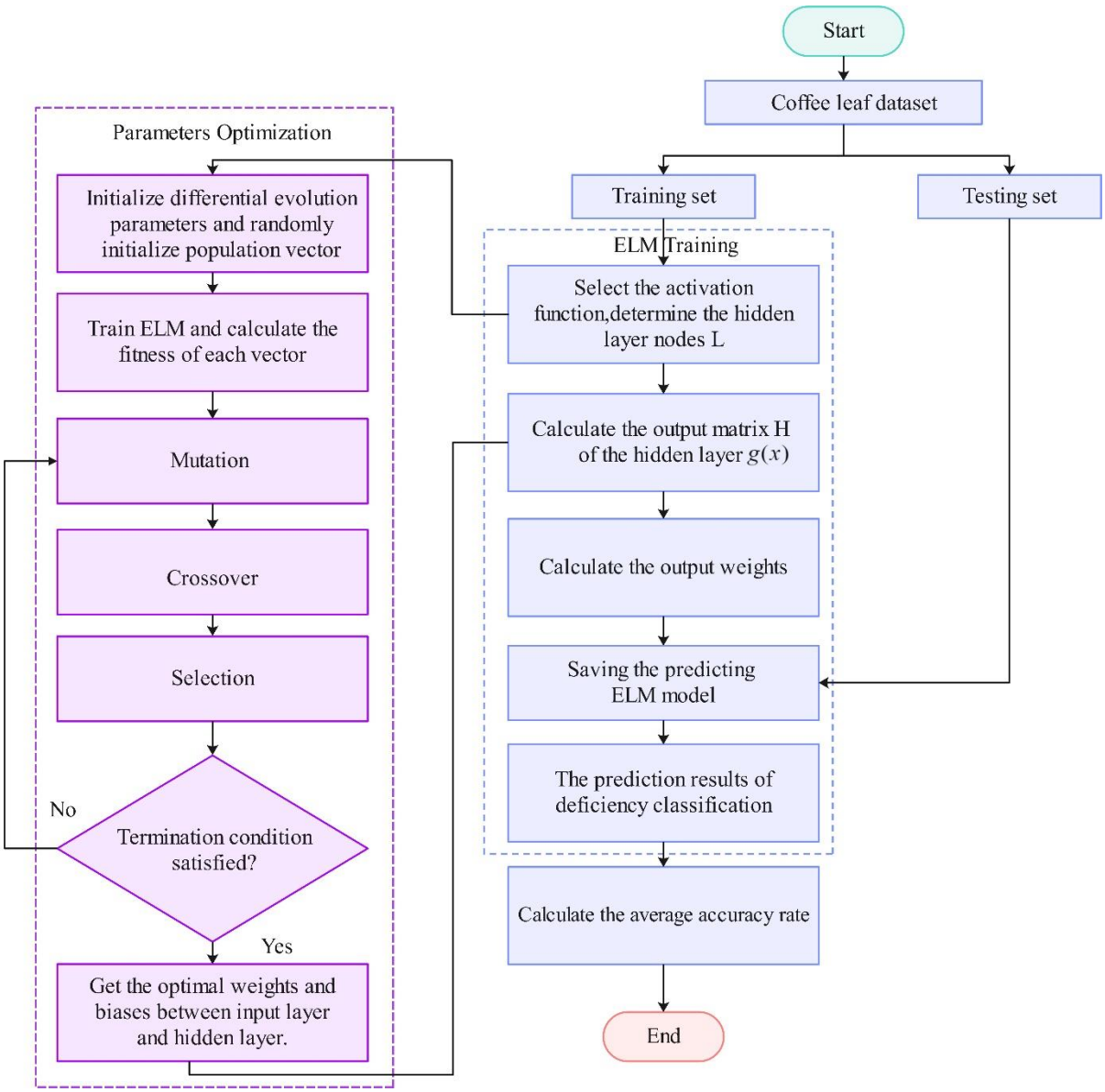


Figure 5- Process flow of the proposed kernel-ELM model

Algorithm 1- Steps of differential evolution algorithm

1: Set population size N
2: Set maximum number of generations G_{max}
3: Population [1 ... N] ← Initialize the population of solutions
4: Best solution ← Null
5: Evaluate fitness value of solutions
6: **For** generation = 1 to G_{max} **do**
7: **For** solution = 1 to N **do**
8: Randomly choose three solutions [$X_{r1,G}$, $X_{r2,G}$, $X_{r3,G}$]
9: Generate mutant solutions ($X_{r1,G}$, $X_{r2,G}$, $X_{r3,G}$)
10: Perform crossover operation
11: Trial solution ← crossover (solution, mutant)
12: **If** fitness of trial solution < fitness of solution **then**
13: Solution ← trial solution
14: **End if**
15: **End for**
16: Current best ← Get best solution from population
17: **If** best solution is null (or) Fitness of current best < Fitness of best solution **then**
18: Best solution ← current best
19: **End if**
20: **End for**

To sum up, the ReLU function is used for each convolutional layer as the nonlinear activation function, while the softmax is used as the activation function. The softmax operates between the output layer and the final dense layer. A $(2,2)$ pool size is utilized. Additionally, the differential evolution algorithm, employed to improve the parameters of the ELM, is outlined as follows (Belciug 2022).

Input: number of iterations G, image dataset D, number of solutions N, $X_i = (\lambda_i, nH_i, fw_i, fh_i, Cp_i, F_i)$ $i = 1, 2, \ldots, N$ the candidate solutions.

Initialization: Randomly generate a set of candidate solutions $X_{i,1}, i = 1, 2, \ldots, N$, and build N networks having λ_i number of convolutional layers and pooling layers, nH_i number of hidden units per convolutional layer, the filter size (fw_i, fh_i), mutation variation factor F_i, and the recombination probability Cp_i.

The networks are trained. The accuracy and loss are recorded. The fitness value of the candidate solution is represented by the loss of each network.

Mutation: The mutation is performed for each individual using the variation factor F_i.

Recombination: For each pair of individuals, perform recombination with Cp_{i}.

Select: The individuals that will be selected for the next generation based on their validation loss.

Repeat: Until the stopping criterion is met (number G of generations is reached).

The steps of the differential evolution algorithm are given in Algorithm 1. Figure 5 shows the process flow of the proposed kernel-ELM model.

Table 3- Details of the Coleaf-DB dataset

Sl. No.	*Nutritional deficiency*	*Quantity*
1	Healthy	6
2	Nitrogen	64
3	Phosphorus	246
4	Potassium	96
5	Magnesium	79
6	Boron	101
7	Manganese	83
8	Calcium	162
9	Iron	65
10	Two or more deficiencies	104

3. Results and Discussion

3.1. Description of dataset

The Coleaf-DB dataset (Tuesta-Monteza et al. 2023), outlined in Table 3, comprises 10 folders representing different nutritional deficiencies: Iron (Fe), Calcium (Ca), Manganese (Mn), Boron (B), Magnesium (Mg), Potassium (K), Phosphorus (P), Nitrogen (N), healthy leaves, and leaves with multiple deficiencies. Each folder contains images annotated with the corresponding nutritional deficiency in their file names. This dataset consists of images in compressed jpeg format with dimensions of 3000×4000 pixels, with a colour depth of 24 bits and vertical and horizontal resolution of 180 dpi.

A healthy leaf image is depicted in Figure 6(a), while the representative characteristics of nutritional deficiencies are described below:

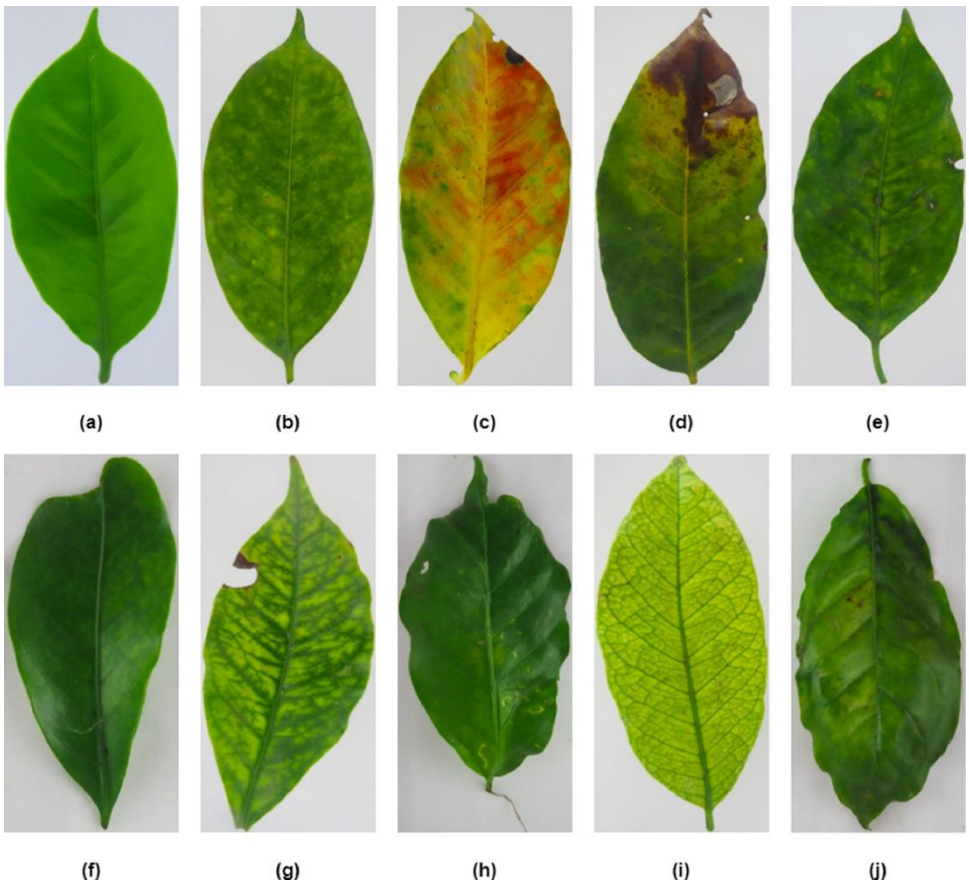


Figure 6- Sample images of Coleaf-DB dataset

1. Nitrogen deficiency: Initial symptoms manifest as uniform chlorosis progressing from the leaf base to the tip. As the deficiency worsens, chlorosis intensifies, covering the whole leaf area. This dataset comprises 64 processed images of leaves with Nitrogen deficiency, as shown in Figure 6(b).

2. Phosphorus deficiency: Old leave exhibits lobular interveinal chlorosis with a reddish hue and yellow-tan spots. There are 246 leaf images for this deficiency, a sample is shown in Figure 6(c).

3. Potassium deficiency: Characterized by yellow-brown spots that later become necrotic with a dark brown shade, often accompanied by curled leaf edges. There are 96 leaf images for Potassium deficiency, a sample is shown in Figure 6(d).

4. Magnesium deficiency: Older leaves show interveinal chlorosis spreading from the base to the tip, often accompanied by green stripes along the midrib. Rapid defoliation may occur. A total of 79 processed images were included for magnesium deficiency, as shown in Figure 6(e).

5. Boron deficiency: Young leaves exhibit small, twisted, and wrinkled characteristics developing from the tip to the base. There are 101 leaf images for boron deficiency. A sample is shown in Figure 6(f).

6. Manganese deficiency: Young leaves display pale green coloration, with deep green main veins and side bands. Leaves may turn increasingly yellow as the deficiency progresses. A total of 83 images were processed, as shown in Figure 6(g).

7. Calcium deficiency: Manifests as marginal chlorosis in new leaves, often associated with leaf deformation and cork formation in the veins. A total of 162 processed images were included, depicted in Figure 6(h).

8. Iron deficiency: New leaves exhibit coloration changing from greenish yellow to nearly white, forming a net-like pattern, while the veins remain green. A total of 65 processed images were included, as shown in Figure 6(i).

From the dataset, 80% of images were used for training, while 20% were used for evaluation. We roughly divided the dataset into two sets: 800 images were assigned to the training set, and 200 images were assigned to the testing set. To accommodate the model input, every image was scaled to a resolution of 224×224 pixels, with pixel normalization applied across all images.

3.2. Implementation details

In this study, we employed Keras to construct the network model, with TensorFlow as the backend. Training hyperparameters used for optimizing the proposed deep learning model are presented in Table 4. These hyperparameters are very important to determine model training efficiency the convergence rate and overall performance of the model. ADAM (Adaptive Moment Estimation) optimizer is chosen since it dynamically adapts learning rates for each parameter and is less likely to diverge or show unstable behavior during training. The batch size is 20, which means before updating the model's weights, 20 training samples are processed. We use a total of 200 epochs so that the model can learn good feature representations for enough number of iterations. The value of the initial learning rate is set to 0.001, which strikes a balance between the convergence speed and the stability of the model, avoiding the risk of overshooting the minima. Moreover, the momentum is set to 0.01, which aids in speeding up the training process by calculating the moving average of the past gradients and preventing oscillations while optimizing. To improve the training process, a variety of augmented images were utilized. The entire algorithm was executed on a workstation running Windows 11, equipped with an Intel CORE i7 12700 h CPU @ 2.4 GHz, and a laptop RTX 3060 GPU.

Table 4- Hyperparameters for the proposed model

Hyperparameter	***Value***
Optimizer	ADAM
Batch size	20
No. of epochs	200
Initial learning rate	0.001
Momentum	0.01

3.3. Performance metrics

To assess the efficacy of the proposed model in classification tasks, metrics are employed to gauge its performance. In this study, a range of metrics is utilized to evaluate the model's ability to identify different deficiencies accurately. The metrics include accuracy, precision, recall, specificity, F1-score, and Matthews correlation coefficient, which provide a complete picture of the classification performance of the model. Mathematical expressions of key classification metrics are presented in Table 5 that are used to assess the performance of the proposed model. These metrics are quantitative measures of how accurate the model is in detecting and classifying nutritional deficiencies in coffee plants. Accuracy is how accurate the model is in its overall correctness, by calculating how many of the samples have been correctly classified out of the total dataset. The second index is Recall (Sensitivity), which reflects the model's power to recognize truly positive cases, to prevent the misclassification of true deficiency samples. Specificity tells us how well we can distinguish negative cases to reduce false positives. Precision is the proportion of correctly classified positive cases out of all predicted positives, which measures how good the positive classifications are. The F1 score is the harmonic mean of precision and recall and is used to balance the two metrics, which is especially important in imbalanced datasets. A robust measure of classification performance is the Matthews Correlation Coefficient (MCC) which considers all possible classification outcomes and hence is particularly useful in situations where there are uneven class distributions, and the model reliability needs to be assessed.

Table 5- Mathematical expression for performance metrics

S. No.	***Performance metric***	***Formula***
1	Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
2	Recall or Sensitivity	$\frac{TP}{TP+FN} \times 100$
3	Specificity	$\frac{TN}{TN+FP}$
4	Precision	$\frac{TP}{TP+FP}$
5	F1-Score	$Precision \times Recall$
6	Mathew Correlation Coefficient (MCC)	$\frac{(TP+TN)}{\sqrt{(TP+FP)+(TN+FN)+(TN\times FP)+(TP\times FN)}}$

Note: TP – True positive, TN – True negative, FP – False positive, FN – False negative.

3.4. Performance comparison

We perform the performance evaluation of the proposed model using different deep learning architectures including well-known baseline models like AlexNet, ResNet-18, ShuffleNet-V2, SqueezeNet, and DenseNet-201 as well as models which have been tested on this dataset before, ResNet-50, Xception, Inception-V3, and MobileNet-V2. Table 6 shows the classification accuracy, precision, recall, specificity, F1-score, and MCC for each model.

Table 6- Performance comparison of the proposed model against other deep learning models

Model	*Accuracy*	*Precision*	*Recall*	*Specificity*	*F1-Score*	*MCC*
AlexNet (Krizhevsky et al. 2017)	95.21	94.86	95.33	0.9674	0.9594	0.9707
ResNet-18 (He et al. 2016)	96.54	96.13	96.66	0.9766	0.9669	0.9691
ShuffleNet-V2 (Ma et al. 2018)	95.54	96.07	95.87	0.9643	0.9532	0.9637
SqueezeNet (Iandola et al. 2016)	93.27	93.94	93.86	0.9459	0.9375	0.9759
DenseNet-201 (Wang & Zhang, 2020)	95.62	96.36	95.11	0.9703	0.9483	0.9735
ResNet-50 (Bera et al. 2024)	97.00	89.00	89.00	-	0.8900	-
Xception (Bera et al. 2024)	98.67	90.00	90.00	-	0.9000	-
Inception-V3 (Bera et al. 2024)	98.67	89.00	89.00	-	0.8900	-
MobileNet-V2 (Bera et al. 2024)	98.33	90.00	89.00	-	0.8900	-
Proposed model	99.50	99.22	99.24	0.9947	0.9957	0.9908

The proposed model has an accuracy of 99.50%, which is superior to all baseline models. It was observed that the proposed model provides better classification performance compared to ResNet-50 (97.00%), Xception (98.67%), Inception-V3 (98.67%), and MobileNet-V2 (98.33%), indicating its effectiveness in identifying various nutritional deficiencies for coffee plants. The multi-model feature extraction strategy, attention-based feature fusion with SE blocks, and optimized Kernel ELM classifier with differential evolution optimization are the reasons for the accuracy improvement in the proposed model. Besides accuracy, the proposed model also performed very well in all other important metrics. With a precision of 99.22% and recall of 99.24%, the model correctly identified deficient leaves with fewer false positives than any other model. The model achieves high specificity (0.9947), meaning that it is very good at distinguishing healthy leaves from deficient leaves and unlikely to misclassify them. Additionally, the F1 score of 0.9957 proves a good balance between precision and recall, indicating that the model is capable of dealing with class imbalances. Finally, the Matthews correlation coefficient of 0.9908 also proves the consistency of the model's prediction, implying that it remains highly reliable even for all the possible classification outcomes.

Additional insight into the misclassification patterns is given by the confusion matrix (Figure 7). This reveals that one sample deficient in Magnesium was misclassified as having multiple deficiencies, which could be due to similar visual characteristics of deficiencies leading to occasional misclassification. Despite this minor limitation, it suggests that future improvement could be possible by using a more sophisticated feature selection technique or a class-specific weighting strategy to improve classification differentiation.

The proposed model was compared with the baseline architectures, such as AlexNet (95.21%), ShuffleNet-V2 (95.54%), and DenseNet-201 (95.62%), which had lower accuracy because of their lower feature extraction capacity as compared to the multi-model fusion approach. However, unlike other baselines, the proposed model outperformed ResNet-18 (96.54%) and ResNet-50 (97.00%) because of their reliance on single feature extraction but not multi-model integration. But the Xception (98.67%), Inception-V3 (98.67%), and MobileNet-V2 (98.33%) models performed well, they didn't have the feature enhancement mechanism based on attention that played a key role in boosting the effectiveness of the proposed approach.

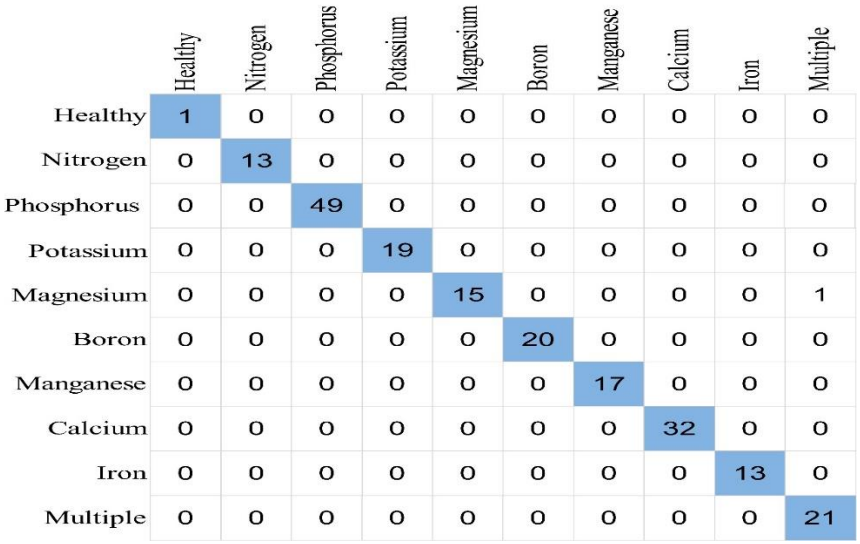


Figure 7- Confusion matrix for the proposed coffee leaf classification model

3.5. Ablation analysis

An ablation study was carried out to further evaluate the contribution of each component in the proposed model by selectively removing or modifying some key modules and analysing the impact on classification performance. The analysis was carried out on three main aspects: effectiveness of attention-based feature fusion with SE blocks, classifier performance of Kernel ELM over other classifiers, and effect of differential evolution optimization on model performance.

3.5.1. Impact of attention-based feature fusion

The first part of the ablation analysis was to study the effect of the attention-based feature fusion using SE blocks. A comparative evaluation between a baseline model using the direct concatenation of features extracted from AlexNet, DenseNet-201, and MobileNet-V2, and the proposed approach where SE-based recalibration is performed before feature fusion is presented. The results shown in Table 7 indicate that the model without SE blocks achieved 96.23% accuracy and with SE blocks 99.50% accuracy. Other performance metrics also showed similar trends, and the proposed model achieved a precision of 99.22% and an F1-score of 0.9957% while the baseline approach yielded a 95.88% precision and 95.82% F1-score. The results demonstrate that SE blocks can improve feature representations by dynamically tuning the importance of different channels and achieving better classification performance.

Table 7- Performance comparison of feature fusion with and without SE blocks

Model Configuration	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*Specificity*	*F1-Score*	*MCC*
Without SE Blocks	96.23	95.88	95.76	0.9541	0.9582	0.9533
With SE Blocks	99.50	99.22	99.24	0.9947	0.9957	0.9908

3.5.2. Impact of Kernel ELM vs. other classifiers

The second part of the analysis is related to the classification performance of Kernel ELM in comparison to other popular classifiers such as Support Vector Machine (SVM) and Random Forest (RF). Evaluation results showed that Kernel ELM was significantly superior to both SVM and RF. The accuracy of Kernel ELM was recorded as 99.50% as shown in Table 8, while SVM and RF were recorded as 92.84% and 94.12%, respectively. Similar to precision and recall, Kernel ELM was more effective in capturing the complex, non-linear patterns in high-dimensional feature space. It is shown that Kernel ELM is more suitable for detecting nutritional deficiencies in coffee leaves, as the substantial improvement of the classification accuracy is confirmed.

Table 8- Classification performance of Kernel ELM compared to other classifiers

Classifier	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*Specificity*	*F1-Score*	*MCC*
SVM	92.84	92.37	91.98	0.91.87	0.9217	0.9232
RF	94.12	94.09	94.01	0.9421	0.9505	0.9414
Kernel ELM	99.50	99.22	99.24	0.9947	0.9957	0.9908

3.5.3. Effectiveness of differential evolution optimization

The third aspect of the ablation study investigated the impact of using DE optimization on the classification performance of Kernel ELM. Kernel ELM was evaluated with and without hyperparameter tuning based on differential evolution. The results as shown in Table 9 indicate that Kernel ELM without differential evolution optimization performed with an accuracy of 97.86% while the proposed model with differential evolution optimization resulted in an accuracy of 99.50%. The observed improvement indicates that the differential evolution algorithm efficiently optimizes the classifier's hyperparameters and improves its generalization power to other types of deficiency classes. Differential evolution optimization was applied to fine-tune kernel parameters to achieve better classification efficiency and to enhance model stability and robustness.

Table 9- Impact of differential evolution optimization on Kernel ELM performance

Model Configuration	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*Specificity*	*F1-Score*	*MCC*
Kernel ELM (Without DE)	97.86	97.61	97.45	0.9730	0.9803	0.9752
Kernel ELM (With DE)	99.50	99.22	99.24	0.9947	0.9957	0.9908

This ablation study demonstrates the necessity of each component in the proposed framework. The addition of SE blocks in feature fusion improved feature representation and classification accuracy. Kernel ELM was used to achieve better classification capability than conventional classifiers and differential evolution optimization was employed to choose optimal parameters for the model. These two components in combination produced a highly accurate and efficient model for the detection and classification of nutritional deficiencies in coffee plants.

3.6. Time complexity analysis

The analysis of the time complexity deals with determining the time taken for the model building for training the proposed model and comparing it with other deep learning architectures. This is important for computational resource usage, training efficiency, and overall feasibility for real-world deployment. However, the training time of the model is a key issue, especially in agricultural applications where we need the model to learn quickly and adapt rapidly.

The average training time of the proposed model is 24.17 minutes which is higher than other deep learning models as shown in Table 10, e.g. AlexNet (12.98 minutes), ResNet-18 (9.89 minutes), MobileNet-V2 (13.75 minutes), and DenseNet-201 (12.43 minutes). This is due to the combination of multiple feature extraction models (AlexNet, DenseNet-201, MobileNet-V2), squeeze-and-excitation-based feature fusion, and differential evolution optimization for Kernel ELM. The combination of the above components helps to achieve better feature representation and classification performance, which in turn results in better accuracy at the expense of increased training time.

The proposed model takes 24.17 minutes to train while the models such as ShuffleNet-V2 (15.18 minutes) and SqueezeNet (14.72 minutes) take less time to train, but do not achieve the same classification accuracy. An increase in computational complexity is justified by a significant improvement in classification performance. Moreover, the DE-based optimization contributes to the computational cost at the same time by fine-tuning the Kernel ELM hyperparameters to achieve a trade-off between model complexity and classification efficiency. Although the training time is slightly higher, the proposed model still trains within an acceptable and practical time for training agricultural problems in reality.

Table 10- Time complexity analysis

Model	*Time (minutes)*
AlexNet	12.98
ResNet-18	9.89
ShuffleNet-V2	15.18
SqueezeNet	14.72
DenseNet-201	12.43
Inception-V3	16.39
MobileNet-V2	13.75
Proposed model	24.17

3.7. Advantages and limitations of the study

The proposed deep learning model for the detection and classification of nutritional deficiencies in coffee plants has a few advantages. The key strength of the proposed model is its performance which is 99.50% accuracy, 99.22% precision, and 0.9957 F1-score, which is better than several benchmark deep learning models. This is due to the attention-based feature fusion mechanism using the squeeze and excitation blocks that dynamically boost the most informative features and suppress the least informative ones. Furthermore, the integration of Kernel ELM with differential evolution optimization has improved the classification efficiency by optimizing hyperparameters without iterative and computationally expensive tuning. This study also

has another major advantage in that we use a multi-model feature extraction strategy: DenseNet-201, AlexNet, and MobileNet-V2 are complementary, they can capture features at different levels of abstraction, which helps to generalize and be robust for real applications.

However, the study has certain limitations that should be recognized. The limitation is the size of the dataset, and the model was trained on a benchmark dataset that although diverse, may not cover all the leaf variations that exist across coffee-growing regions. However, this may lead to a reduction in the generalizability of the model in the case of new datasets from different environmental conditions. The second limitation is computational complexity, especially in the training phase. The model uses MobileNet-V2 for lightweight computation, however, the combination of several deep learning architectures and ELM leads to relatively higher computational cost than single model-based methods. Furthermore, the study does not take into consideration the real-time deployment aspects like model integration into a mobile app or an edge computing system to diagnose the coffee farms on site.

4. Conclusions

This study proposed a deep learning-based approach for the detection and classification of nutritional deficiencies in coffee plants. DenseNet-201, AlexNet, and MobileNet-V2 were integrated into the model for multi-scale feature extraction, and an attention-based feature fusion mechanism was used to fuse features based on squeeze-and-excitation blocks to extract more discriminative features. The Kernel ELM optimized with a differential evolution algorithm was used for the classification, which is efficient and accurate. The benchmark dataset was used to evaluate the proposed model and compare it to baseline and state-of-the-art deep learning models in terms of identifying different types of nutritional deficiencies. In the study, it was demonstrated that multi-model feature extraction, attention-based feature fusion, and an optimization algorithm should be integrated to enhance classification performance in agricultural applications.

Experimental results validated that the proposed model was superior to existing deep learning approaches in terms of accuracy and efficiency. This study's key findings are as follows:

- We demonstrate that the proposed multi-model feature extraction strategy is effective in capturing a diverse set of feature representations that enhance classification robustness.

- Feature importance is enhanced by the attention-based feature fusion mechanism using squeeze-and-excitation blocks, which results in better classification accuracy than the traditional feature concatenation methods.

- Significant reduction in computational cost is achieved without significantly compromising classification accuracy by using the Kernel ELM classifier optimized by the differential evolution algorithm.

- The model predicts with a classification accuracy of 99.50%, precision of 99.22%, recall of 99.24%, F1-score of 0.9957%, and an MCC of 0.9908, and it outperforms existing deep learning models.

The proposed model has shown superior classification performance; however, future research should be focused on generalizing and scaling its performance. Also, the dataset can be expanded to a larger set of more diverse leaf samples from different environmental conditions to check the robustness of the model. Furthermore, the model can be applied on edge computing devices or mobile platforms in order to detect deficiency in real time under field conditions. Additionally, lightweight architectures, model pruning techniques, and self-supervised learning techniques could be further explored to improve computational efficiency with accuracy. The proposed approach can be further optimized to support precision agriculture and sustainable coffee cultivation by addressing these areas.

References

Abuhayi B M & Mossa A A (2023). Coffee disease classification using Convolutional Neural Network based on feature concatenation. *Informatics in Medicine Unlocked*, 39: 101245. doi.org/10.1016/j.imu.2023.101245

Adelaja O & Pranggono B (2025). Leveraging Deep Learning for Real-Time Coffee Leaf Disease Identification. *AgriEngineering* 2025, 7, 13. doi.org/10.3390/agriengineering7010013

Ali A M, Słowik A, Hezam I M & Abdel-Basset M (2024). Sustainable smart system for vegetables plant disease detection: Four vegetable case studies. *Computers and Electronics in Agriculture*, 227, 109672. doi.org/10.1016/j.compag.2024.109672

Apraez L S C, Pino A F S, Ossa A, Vasquez C I, Solarte J D, Cabrera E V R, & Ruiz S E (2025). Application of Spectral Imaging and Vegetation Index in Latin American Coffee Production: A Systematic Mapping. *Land Degradation & Development*, 36(2), 337-349. doi.org/10.1002/ldr.5373

Baiju B V, Kirupanithi N, Srinivasan S, Kapoor A, Mathivanan S K, & Shah M A (2025). Robust CRW crops leaf disease detection and classification in agriculture using hybrid deep learning models. *Plant Methods*, 21, 18. doi.org/10.1186/s13007-025-01332-5

Barman U, Sarma P, Rahman M, Deka V, Lahkar S, Sharma V, & Saikia M J (2024). Vit-SmartAgri: vision transformer and smartphone-based plant disease detection for smart agriculture. Agronomy, 14(2), 327. doi.org/10.3390/agronomy14020327

Belciug S (2022). Learning deep neural networks' architectures using differential evolution. Case study: medical imaging processing. Computers in biology and medicine 146: 105623. doi.org/10.1016/j.compbiomed.2022.105623

Bera A, Bhattacharjee D & Krejcar O (2024). PND-Net: plant nutrition deficiency and disease classification using graph convolutional network. *Scientific Reports*, 14(1), 15537. doi.org/10.1038/s41598-024-66543-7

Bhagat M & Kumar D (2023). Efficient feature selection using BoWs and SURF method for leaf disease identification. *Multimedia Tools and Applications* 82: 28187–28211. doi.org/10.1007/s11042-023-14625-5

Bidarakundi P M & Kumar B M (2024). Coffee-Net: Deep Mobile Patch Generation Network for Coffee Leaf Disease Classification. *IEEE Sensors Journal*. doi.org/10.1109/jsen.2024.3498050

Chug, A, Bhatia A, Singh A P & Singh D (2023). A novel framework for image-based plant disease detection using hybrid deep learning approach. *Soft Computing*, 27(18), 13613-13638. doi.org/10.1007/s00500-022-07177-7

de Oliveira Aparecido L E, Lorençone P A, Lorençone J A, Torsoni G B, de Lima R F, Padilha F & de Souza Rolim G (2024). Addressing coffee crop diseases: forecasting Phoma leaf spot with machine learning. *Theoretical and Applied Climatology* 155(3): 2261-2282. doi.org/10.1007/s00704-023-04739-z

de Souza Dias J, Sorte L X B, Fambrini F & Saito J H (2025, February). Coffee plant disease detection using JSEG segmentation and near sets clustering. In Fifth Symposium on Pattern Recognition and Applications (SPRA 2024) (Vol. 13540, pp. 40-52). SPIE. doi.org/10.1117/12.3056434

Demilie W B (2024). Plant disease detection and classification techniques: a comparative study of the performances. *Journal of Big Data* 11(1): 5. doi.org/10.1186/s40537-023-00863-9

Dosset A, Dang L M, Alharbi F, Habib S, Alam N, Park H Y & Moon H (2025). Cassava disease detection using a lightweight modified soft attention network. *Pest Management Science*, 81(2), 607-617. doi.org/10.1002/ps.8456

Duhan S, Gulia P, Gill N S & Narwal E (2025). RTR_Lite_MobileNetV2: A Lightweight and Efficient Model for Plant Disease Detection and Classification. *Current Plant Biology*, 100459. doi.org/10.1016/j.cpb.2025.100459

Faisal M, Leu J S & Darmawan J T (2023). Model selection of hybrid feature fusion for coffee leaf disease classification. *IEEE Access*, 11: 62281-62291. doi.org/10.1109/access.2023.3286935

Haridasan A, Thomas J, & Raj E D (2023). Deep learning system for paddy plant disease detection and classification. *Environmental monitoring and assessment*, 195(1), 120. doi.org/10.1007/s10661-022-10656-x

Hasan R I, Yusuf S M, Mohd Rahim M S & Alzubaidi L (2023). Automatic clustering and classification of coffee leaf diseases based on an extended kernel density estimation approach. *Plants*, 12(8), 1603. doi.org/10.3390/plants12081603

He K, Zhang X, Ren S & Sun J (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778). doi.org/10.1109/cvpr.2016.90

Hitimana E, Sinayobye O J, Ufitinema J C, Mukamugema J, Rwibasira P, Murangira T & Ngabonziza J (2023). An intelligent system-based coffee plant leaf disease recognition using deep learning techniques on Rwandan Arabica dataset. *Technologies*, 11(5), 116. doi.org/10.3390/technologies11050116

Iandola F N, Han S, Moskewicz M W, Ashraf K, Dally W J & Keutzer K (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360.

Idress K A D, Gadalla O A A, Öztekin Y B & Baitu G P (2024). Machine Learning-based for Automatic Detection of Corn-Plant Diseases Using Image Processing. *Journal of Agricultural Sciences (Tarim Bilimleri Dergisi)*, 30(3), 464-476. doi.org/10.15832/ankutbd.1288298

Karia A J, Ally J S & Leonard S (2025). Enhancing Coffee Leaf Rust Detection Using DenseNet201: A Comprehensive Analysis of the Mbozi and Public Datasets in Songwe, Tanzania: English. *African Journal of Empirical Research*, 6(1), 171-188. doi.org/10.51867/ajernet.6.1.17

Karthik R, Alfred J J & Kennedy J J (2023). Inception-based global context attention network for the classification of coffee leaf diseases. *Ecological Informatics*, 77, 102213. doi.org/10.1016/j.ecoinf.2023.102213

Krizhevsky A, Sutskever I & Hinton G E (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90.. doi.org/10.1145/3065386

Ma N, Zhang X, Zheng H T & Sun J (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European conference on computer vision (ECCV) (pp. 116-131).

Milke E B, Gebiremariam M T & Salau A O (2023). Development of a coffee wilt disease identification model using deep learning. *Informatics in Medicine Unlocked*, 42, 101344. doi.org/10.1016/j.imu.2023.101344

Nawaz M, Nazir T, Javed A, Amin S T, Jeribi F & Tahir A (2024). CoffeeNet: A deep learning approach for coffee plant leaves diseases recognition. *Expert Systems with Applications*, 237, 121481. doi.org/10.1016/j.eswa.2023.121481

Pham T C, Le C H, Packianather M & Hoang V D (2023, December). Artificial intelligence-based solutions for coffee leaf disease classification. *In IOP Conference Series: Earth and Environmental Science* (Vol. 1278, No. 1, p. 012004). IOP Publishing. doi.org/10.1088/1755-1315/1278/1/012004

Pillaca-Pullo O S, Moreni Lopes A, Bautista-Cruz N & Estela-Escalante W (2025). From coffee waste to nutritional gold: bioreactor cultivation of single-cell protein from Candida sorboxylosa. *Journal of Chemical Technology & Biotechnology*, 100(2), 360-368. doi.org/10.1002/jctb.7778

Preethi P, Swathika R, Kaliraj S, Premkumar R & Yogapriya J (2024). Deep Learning–Based Enhanced Optimization for Automated Rice Plant Disease Detection and Classification. *Food and Energy Security*, 13(5), e70001. doi.org/10.1002/fes3.70001

Raghavendra B K (2023). An Efficient Approach for Coffee Leaf Disease Classification and Severity Prediction. *International Journal of Intelligent Engineering & Systems*, 16(5). doi.org/10.22266/ijies2023.1031.59

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

Ramamurthy K, Thekkath R D, Batra S & Chattopadhyay S (2023). A novel deep learning architecture for disease classification in Arabica coffee plants. *Concurrency and Computation: Practice and Experience*, 35(8), e7625. doi.org/10.1002/cpe.7625

Rezaei M, Diepeveen D, Laga H, Gupta S, Jones M G & Sohel F (2025). A transformer-based few-shot learning pipeline for barley disease detection from field-collected imagery. *Computers and electronics in agriculture*, 229, 109751. doi.org/10.1016/j.compag.2024.109751

Salamai A A (2024). Towards automated, efficient, and interpretable diagnosis coffee leaf disease: A dual-path visual transformer network. *Expert Systems with Applications*, 255, 124490. doi.org/10.1016/j.eswa.2024.124490

Sandhiya B & Raja S K S (2024). Deep learning and optimized learning machine for brain tumor classification. *Biomedical Signal Processing and Control*, 89, 105778. doi.org/10.1016/j.bspc.2023.105778

Sandler M Howard A, Zhu M, Zhmoginov A & Chen L C (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018, 4510–4520 (Computer Vision Foundation/IEEE Computer Society, 2018). doi.org/10.1109/cvpr.2018.00474

Sankareshwaran S P, Jayaraman G, Muthukumar P & Krishnan A (2023). Optimizing rice plant disease detection with crossover boosted artificial hummingbird algorithm based AX-RetinaNet. *Environmental Monitoring and Assessment*, 195(9), 1070. doi.org/10.1007/s10661-023-11612-z

Saygılı A (2023). The efficiency of transfer learning and data augmentation in lemon leaf image classification. *European Journal of Engineering and Applied Sciences* 6(1): 32-40. doi.org/10.55581/ejeas.1321042

Shovon M S H, Mozumder S J, Pal O K, Mridha M F, Asai N & Shin J (2023). PlantDet: A robust multi-model ensemble method based on deep learning for plant disease detection. *IEEE Access*, 11, 34846-34859. doi.org/10.1109/access.2023.3264835

Shrma V S T & Rahiman M K (2024). A comparison of CNN and SVM algorithms for the prediction of growth defects in coffee plants for stable yield and fungal diseases. *Interactions*, 245(1), 322. doi.org/10.1007/s10751-024-02135-1

Singh M K & Kumar A (2024). Coffee Leaf Disease Classification by Using a Hybrid Deep Convolution Neural Network. *SN Computer Science*, 5(5), 618. doi.org/10.1007/s42979-024-02960-9

Sofuoğlu C İ & Bırant D (2023). Using convolutional neural network for grape plant disease classification. Uludağ Üniversitesi Mühendislik Fakültesi Dergisi, 28(3), 809-820. doi.org/10.17482/uumfd.1277418

Sofuoğlu C İ & Bırant D (2024). Potato plant leaf disease detection using deep learning method. *Journal of Agricultural Sciences (Tarim Bilimleri Dergisi)*, 30(1), 153-165. doi.org/10.15832/ankutbd.1276722

Thakur D, Gera T, Aggarwal A, Verma M, Kaur M, Singh D & Amoon M (2024). SUNet: Coffee Leaf Disease Detection using Hybrid Deep Learning Model. *IEEE Access*. doi.org/10.1109/access.2024.3476211

Tsigkou K, Demissie B A, Hashim S, Ghofrani-Isfahani P, Thomas R, Mapinga K F & Angelidaki I (2025). Coffee processing waste: Unlocking opportunities for sustainable development. *Renewable and Sustainable Energy Reviews*, 210, 115263. doi.org/10.1016/j.rser.2024.115263

Tuesta-Monteza V A, Mejia-Cabrera H I & Arcila-Diaz J (2023). CoLeaf-DB: Peruvian coffee leaf images dataset for coffee leaf nutritional deficiencies detection and classification. *Data in Brief*, 48, 109226. doi.org/10.1016/j.dib.2023.109226

Upadhyay A, Chandel N S, Singh K P, Chakraborty S K, Nandede B M, Kumar M & Elbeltagi A (2025). Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture. *Artificial Intelligence Review*, 58(3), 1-64. doi.org/10.1007/s10462-024-11100-x

Wang S H & Zhang Y D (2020). DenseNet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(2s), 1-19. doi.org/10.1145/3341095

Wang X & Liu J (2025). TomatoGuard-YOLO: a novel efficient tomato disease detection method. *Frontiers in Plant Science*, 15, 1499278. doi.org/10.3389/fpls.2024.1499278

Wang Y, Yu X & Zhang W (2025). An improved reinforcement learning-based differential evolution algorithm for combined economic and emission dispatch problems. *Engineering Applications of Artificial Intelligence*, 140, 109709. doi.org/10.1016/j.engappai.2024.109709

Yamashita J V Y B & Leite J P R (2023). Coffee disease classification at the edge using deep learning. *Smart Agricultural Technology*, 4, 100183. doi.org/10.1016/j.atech.2023.100183

Yazdani M, Kabirifar K & Haghani M (2024). Optimising post-disaster waste collection by a deep learning-enhanced differential evolution approach. Engineering Applications of Artificial Intelligence, 132, 107932. doi.org/10.1016/j.engappai.2024.107932

Copyright © 2025 The Author(s). This is an open-access article published by Faculty of Agriculture, Ankara University under the terms of the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium or format, provided the original work is properly cited.