SN Computer Science (2024) 5:618
https://doi.org/10.1007/s42979-024-02960-9

ORIGINAL RESEARCH

Coffee Leaf Disease Classification by Using a Hybrid Deep Convolution Neural Network

Manish K. Singh[1] · Avadhesh Kumar[1]

Received: 8 November 2023 / Accepted: 10 May 2024

Abstract
The most common symptoms of coffee leaf disease are coffee leaf rust, black rot diseases, and brown eye spot. Leaf rust is the first symptom of coffee leaves that is visible on the upper surface of the leaves, i.e., small pale-yellow spots. At later stages, these spots gradually expand their diameter and become visible on the undersurface. The underside of the leaves becomes orange–yellow in colour or red–orange powdery, which varies from region to region. Finally, these leaf diseases rapidly spread and result in significant economic losses, so it's essential to identify plant diseases in their initial phases. Detailed features of the leaves have been required for this task to differentiate healthy and unhealthy coffee leaves. The proposed model utilized a mix of two deep learning methods, i.e., InceptionV3 and DenseNet121, and performed better using the concatenation and augmentation approach of convolution neural network (CNN). The proposed work contrasts existing pre-trained CNN models, such as InceptionV3 and DenseNet121, which require more computing power and parameters. According to the proposal, the hybrid deep convolution neural network model demonstrated a remarkable accuracy rate of 99%, which proved better than existing models such as InceptionV3, EfficientNetB0 and DenseNet121, which gained 94%, 98% and 96.36% accuracy, respectively.

Keywords Convolution neural networks (CNNs) · Coffee leaf rust · Black rot disease · Brown eye spot · Data augmentation

Introduction

Coffee is widely regarded as the second-largest industry globally, following crude oil. It is a beneficial agricultural product. Around the world, fifteen billion tree plantations contribute to society. Coffee production serves the needs of around 25 million producers in various countries, catering to global demand. The commercial coffee industry predominantly revolves around two main species: Arabica coffee (*Coffea arabica* L.) and Robusta coffee (*Coffea canephora*). Arabica coffee predominantly flourishes in damp tropical highlands, while Robusta coffee can prosper at sea level but is also commonly cultivated in these moist tropical highlands [1] 0.97.55% of coffee cultivars are from Brazil, which produces 40% of the largest producer. Since 1849, Arabica coffee has thrived in the highlands of Thailand, making it a suitable and viable option As a lucrative crop to substitute for opium production. The cultivation of Arabica coffee in this region proves highly appropriate and feasible, offering a sustainable alternative to opium derived from the dried latex of the opium poppy and Papaver somniferum [2]. The most controversial challenge in the Ethiopian coffee sector is the coffee-producing country's that use of a plant with stimulating characteristics called khat (Catha edulis) instead of coffee for its benefit [3, 4]. Small, low-profile coffee-growing families perform a number of the harvest some famines harm and sicken the plant, decreasing production and affecting the food's quality and flavour [5]. A mobile application was developed using fuzzy inference to diagnose crop disease [6]. A deep neural network model has been used to quantify and categorize coffee leaf rust and leaf miner, which achieved above 97% symptom of classification results, making it possible to evaluate the severity of biotic stress [7, 8]. Small producers needed help to keep a steady supply of premium coffee on the market [9]. Colombia grows only Arabica coffee beans, Compared to Canephora coffees also known as Robusta coffees, these have a softer flavour and

✉ Manish K. Singh
manish5004@gmail.com

Avadhesh Kumar
avadheshkumar@galgotiasuniversity.edu.in

[1] Galgotias University, Greater Noida, India

Published online: 04 June 2024

are more sought after on international markets. Most harvesting is done by medium-sized or smaller families specialising in coffee cultivation as a result of being attacked by pests, plants can lose quality and quantity [10]. The growing severity of poor cultivation has impacted quality and quantity indicators over the past few decades [2]. Coffee leaf rust (CLR) [11], coffee borer beetle (Hypothenemus hampei) [12], coffee leaf miner, citrus mealybug, root borer, red spider and coffee stem mite severely damage coffee plants, affecting crop yield [13, 14]. The high level of horizontal resistance in coffee populations and the tremendous genetic variety of coffee populations may have kept the rust at bay. CLR aggression is height dependent, with more aggressive behaviour at lower elevations and warmer temperatures. Coffee leaf rust stands as the most devastating coffee disease that affect *C. arabica* plants [15, 16]. After the nineteenth century, CLR damaged many coffee crops in large areas of Ceylon's Arabica. Many countries use pesticides to prevent rust, which still causes production declines of 20% or more in some countries [17]. CLR was only found in Ethiopia in 1934, although it had been present in other countries for long periods without spreading epidemics or wiping out specific C. arabica types. Coffee and rust have coexisted for a very long period. CBD was found for the first time in Kenya in 1922, west of the Rift Valley and close to Mount Elgon. In some reports, it was found that it damages the crops by up to 75%. According to this, coffee growing west of the Rift Valley ended, and tea plantations became the dominant form of agriculture in the region. The Rift Valley's arid climate kept the disease from spreading to the Central African country's crucial coffee-growing regions for a long time—the province's highlands. For the first time, Ray-ner noted the presence of CBD in 1951, east of the Rift [18, 19]. Deep learning is a new class of models that have emerged in artificial intelligence due to the tremendous growth in machine learning's practical applications over the past few years. The deep learning approach named Convolution Neural Networks (CNN) has achieved excellent results in image recognition. CNN automatically learns relevant characteristics from the dataset used for training, eliminating the need for manually crafted elements based on prior problem understanding used in traditional methods. Moreover, CNN integrates the segmentation step within its convolution filters, simplifying its usage further [19, 20]. Brazil, the world's top producer of coffee, claims the International Coffee Organisation, which is highly advantageous for the other country. Coffee plantations encounter various biotic stresses like leaf miners, rust, and brown leaf spot. These stresses lead to defoliation, reduced photosynthesis, and, ultimately, a negative impact on crop yield [21]. To organise the information, developed a system a threshold-based method was used to assess the various symptoms and signs associated with coffee leaves rather than categorising them. Leaf miner and rust are the two categories used to classify the symptoms. Both of these categories included components that were made by hand. The segmentation of lesions yielded data that demonstrated the symptoms could be differentiated and rated on a severity scale. While the results were satisfactory, this segmentation technique fails to account for the interplay between pixel positions, which can be altered by lighting and specular reflection factors. The main achievements of this project can be encapsulated as follows: Firstly, the dataset has been expanded by including more coffee leaves, enabling a larger and more diverse collection of images of both healthy and ill leaves. Secondly, the project introduces a multi-task system that performs categorization and severity evaluation of biological stress on coffee leaves. Lastly, the project conducts comparative studies on various deep learning structures and exploring their effectiveness in analyzing the coffee leaf images [22]. An algorithm for pattern recognition based on texture is proposed to calculate features such as local binary features and statistical features to identify a coffee plant leaves with lesions and compare its detection rate with other CNN methods [23]. The CLR disease is the primary phytosanitary threat to coffee plants, and it is caused by the fungus name Hemileia vastatrix. It's not uncommon for the severity of a problem to increase before any significant measures are taken to fix it, Poor disease control can severely weaken coffee plants as a result leaving only a handful of leaves on the trees and dramatically reducing yield and quality [24]. Deep learning techniques are a machine learning subfield that entails training artificial neural networks to identify objects in pictures, transcribe speech, and make predictions. Deep learning models are built from layers of interconnected artificial neurons and can learn to recognise patterns in data through back propagation [25]. In this research, a novel classification model is presented to effectively detect various ailments affecting coffee leaves, like grey leaf spots, common rust, and northern leaf blight, using coffee leaf images. The model utilizes Convolution neural networks (CNNs) with two prior trainings, namely InceptionV3 and DenseNet121, chosen for their reasonable parameter sizes. To enhance the classification performance, feature fusion employed to integrate the forecasting abilities of both CNN models, creating a comprehensive and accurate classification system. The main aim of this study is to utilize feature fusion method in order to combine the predictive capabilities of two convolution neural networks (CNNs), InceptionV3 CNN and DenseNet121 CNN that have already been trained. By implementing an end-to-end learning model, the features derived from both CNNs are fused together. The objective is to enhance the classification accuracy of coffee plant diseases. The key points of this work can be encapsulated as follows:

a. The application of feature fusion in a paradigm of end-to-end learning, combining features extracted from InceptionV3 CNN and DenseNet121 CNN.
b. The evaluation of classification accuracy using the proposed model, as well as comparisons with DensenNet121, EfficientNetB0 and InceptionV3 models, using the same dataset.
c. The achievement of higher accuracy in detecting and classifying coffee plant diseases with the proposed model, while maintaining reasonable parameter sizes.

Related Work

In [26], the authors proposed a model that achieved a classification accuracy of 98.56% by combining traits from multiple CNN approaches. In [27], the author's model achieved an accuracy rate of 98% by addressing the issue of over-fitting through data augmentation and transfer learning. In [28], authors presented a densely optimized CNN to classify four types of maize leaves, including healthy maize leaves and three affected by disease leaves like: corn grey leaf spot, common corn rust, and corn northern leaf blight. The proposed model deployed a softmax classifier layer followed by five dense blocks and Attained a classification accuracy of 94%. In [29], the suggested transfer learning method using DenseNet-121 Attained a rate of classification accuracy of 99.81% and required fewer training parameters, reducing computing complexity. In [30], authors retrieved deep features from tomato, maize, and potato photos stored in the Plant Village dataset repository using the pre-trained VGG19. They concentrated on extracting Information from layers 6 and 7 of the VGG19 network, which was then linked using the partial least squares (PLS) method. They used the partial least squares (PLS) technique, which included fusing the extracted features in parallel to choose the most appropriate features. The network achieved an impressive classification accuracy of 97.50% with these parameters. In [31], The authors suggested a methodology that integrates both classification and metaheuristic techniques for the purpose of categorizing breast cancer. In [32], the authors suggested an optimized approach for detecting breast cancer through the examination of ultrasound images. In [33], a combination of features was utilized by the authors to predict breast cancer, merging patient gene modality data with image data. In [34], a method involving the application of fuzzy-based association rules mining was employed by the authors to analyze data associated with breast cancer. Decision trees were applied to anthropometric and blood data in [35] to identify symptoms of breast cancer, resulting in highly favorable outcomes. Lastly, in [36], a two-stage filter-based method for breast cancer analysis was implemented by the authors, incorporating two statistical methods in the first stage and two theoretical approaches in the second stage.

Materials and Methods

A convolutional neural network (CNN) is comprised of three layers comprising convolution, pooling, and fully connected layers. These layers work together to create a powerful framework for analyzing image datasets. This framework requires minimal preprocessing compared to other classification algorithms, given sufficient training. CNNs capable of learning filters and extracting features from images, enabling them to effectively process large areas. Compared to other neural networks, they are exceptionally good at processing visual, audio, or voice signals [37].

Classification using CNN

Multiple layers of CNN describe as follows:

a) **Input Layer:** This layer provides raw data for a matrix of pixel values.
b) **Convolutional Layer:** This layer conducts convolution operations on the input layer and includes trainable filters aimed at extracting features from the image to assist in pattern recognition.
c) **Activation Layer:** In this layer, the activation function applies for non-linearity in the model that can identify a more complex pattern.
d) **Pooling Layer (Subsampling Layer):** In this layer, with the help of max-pooling and average-pooling, the output's spatial dimensions are diminished thus decreasing the parameter count and computational load.
e) **Fully Connected Layer (FC Layer):** This layer establishes connections between every neuron in the preceding layer and every neuron in the present layer, facilitating the learning of intricate patterns and prediction-making.
f) **Flatten Layer:** This layer forwards the output from the layer below to the fully connected layers.
g) **Dropout Layer:** This layer uses to prevent overfitting and improve generalisation.
h) **Batch Normalisation Layer:** This layer uses to normalise the previous layer's output.

CNNs utilize the softmax function to ascertain the correct class for an image, operating as a probability distribution function. This function assigns probabilities to each potential class, indicating the likelihood of the input image belonging to a specific category. Equation 1 represents the softmax process, where (X)i represent the input value, and N signifies the total number of types. First, to compute the class probability, exponentiated the input values (X)i, then treat any negative values as positive [38].

$$\sigma(x)_i = \frac{e^{X_i}}{\sum_{j=1}^{N} e^{X_j}} for\, i = 1 \ldots\ldots .N\, and\, X = \left(X_1 \ldots\ldots .X_N\right) \tag{1}$$

Feature Extraction Process

To achieve this objective, we employ a sequence of layers to analyze the image and extract pertinent information. Following multiple convolutional and activation layers, pooling layers are implemented. The classification module utilizes the pixel values of the image to generate a feature map. The convolutional layer's filters learn low-level, intermediate, and high-level image features as they apply to the layer's input. Pooling includes edges, patterns, and textures. Typically, A tensor filters data with odd dimensions involved in the information in a sliding-window fashion by convolutional averaging the filter values after multiplying them by the window on the result. Equation 2 represents the convolutional product of tensors T (input) and F (filter), while the height, channel count, and width signify $n_h, n_c and n_w$, respectively.

$$Conv(T,F)_{m,n} = \sum_{x=1}^{p_h} \sum_{y=1}^{p_w} \sum_{z=1}^{p_c} F_{x,y,z} \times T_{m+x-1,n+y-1,z} \tag{2}$$

Equation 3 indicates Rectifier Linear Unit (ReLU) activation function. It acts as a non-linear transform on the result of its convolutional layer [21].

$$F(x) = \max(0, m) \tag{3}$$

The activation layers add non-linearity to the output using the activation function, which researchers have represented in Eq. 4.

$$Conv(T^{[A-1]}, F^{(p)})_{m,n} = \sum_{x=1}^{p_h} \sum_{y=1}^{p_w} \sum_{z=1}^{p_c} (F_{x,y,z} \times T_{m+x-1,n+y-1,z} + b_p^{[A]}) \tag{4}$$

The input layer, shown below, depicts $T^{[A-1]}$ with $p_h, p_w and p_c$ Activation layers, which are typically defined after convolutional layers, introduce nonlinearity to the output. Equation 5 illustrates the computation of the output layer, where $F^{(n)}$ represents the dimensions of the filters.

$$T^A = \left[\psi^{[A]}(conv\left(T^{[A-1]}, F^{(1)}\right), \ldots\ldots.\psi^{[A]}(conv\left(T^{[A-1]}, F^{n_c^A}\right)\right] \tag{5}$$

Classification Process

Classification is employed to assign the collected characteristics to the appropriate categories. In typical scenarios, fully connected layers connect all the elements. In these layers, each input connects to every neuron in the subsequent activation layer. This network configuration functions as a classifier, where each layer consists of numerous neurons connected to the output layer. Equation 6 represents the SoftMax function, which normalises the output by utilising the SoftMax activation function to determine the highest probabilities.

$$\sigma(P)_i = \frac{e^{P_i}}{\sum_{i=1}^{k} e^{P_i}} \tag{6}$$

Transfer Learning Method

One machine learning method, "transfer learning", involves taking a model built for one task and using it as the foundation for another. When employing deep learning for tasks in computer vision or natural language processing, it's customary to initiate the process with pre-trained models. The enormous skill gains provided by neural network models necessitate their usage for addressing problems related to massive computational time resources. Model Development and Pre-training are the two transfer learning strategies that are most frequently used [38].

Essential Points of Model Approach Development

1. Choose a source job with abundant data and a known relationship between the input and output data. The system learned their relationship while mapping the input to the output data.
2. Develop a Competent Model for This Primary Task Next, Construct a Source Model. To qualify as "feature learning," the model must outperform a naive model.
3. Model Reuse: The model may be used whole or in part, depending on the techniques used.
4. Input/Output Data Model: Depending on the input/output data for the intended task, the model might need to be adjusted or improved.

Approach for Pre-Trained Model

1. Select Pre-Trained Source Model-We have used InceptionV3 and DenseNet121 as pre-trained models for this work.

2. Building a new model based on a previously trained model is a method used frequently in machine learning known as transfer learning. Using another model can enhance the knowledge gained by a separate but related model in transfer learning.
3. Tune Model: Taking an already trained model and further training it on a new dataset related to the original task allows for fine-tuning the model.
4. It would be taught how to map the feature extractor's output classes to the new data set. The InceptionV3 and DenseNet121 architectures of convolutional neural networks were utilized in this work, both of which had been pre-trained. In Fig. 1, InceptionV3, part of Google's Inception architectural lineage, employs diverse inception modules featuring various filter sizes to capture information across different scales. Designed for discerning intricate image patterns, the model combines these modules with traditional convolutional layers to gather features at multiple scales. Although it demands more parameters and increases computational costs, InceptionV3 addresses this by incorporating additional classifiers during training to enhance gradient flow and convergence. Recognized for its versatility, InceptionV3 is beneficial for various computer vision tasks, encompassing feature extraction, object recognition, and image classification [43].

An initial convolutional layer followed by four dense blocks and three transition layers represents the structure of DenseNet121. In this structure, each dense block densely connects several thick layers. The purpose of transition layers is to decrease the number of maps with features and to down-sample the spatial dimensions of the input. After the final dense block is processed, as Fig. 2 illustrates, a pooling layer and softmax activation employed to acquire feature maps and generate final class probabilities. In deep neural networks, particularly in architectures like DenseNet, it is common to use a substantial block consisting of six dense layers. Every thick layer comprises a 1 × 1 convolutional layer, then a 3 × 3 convolutional layer. This configuration

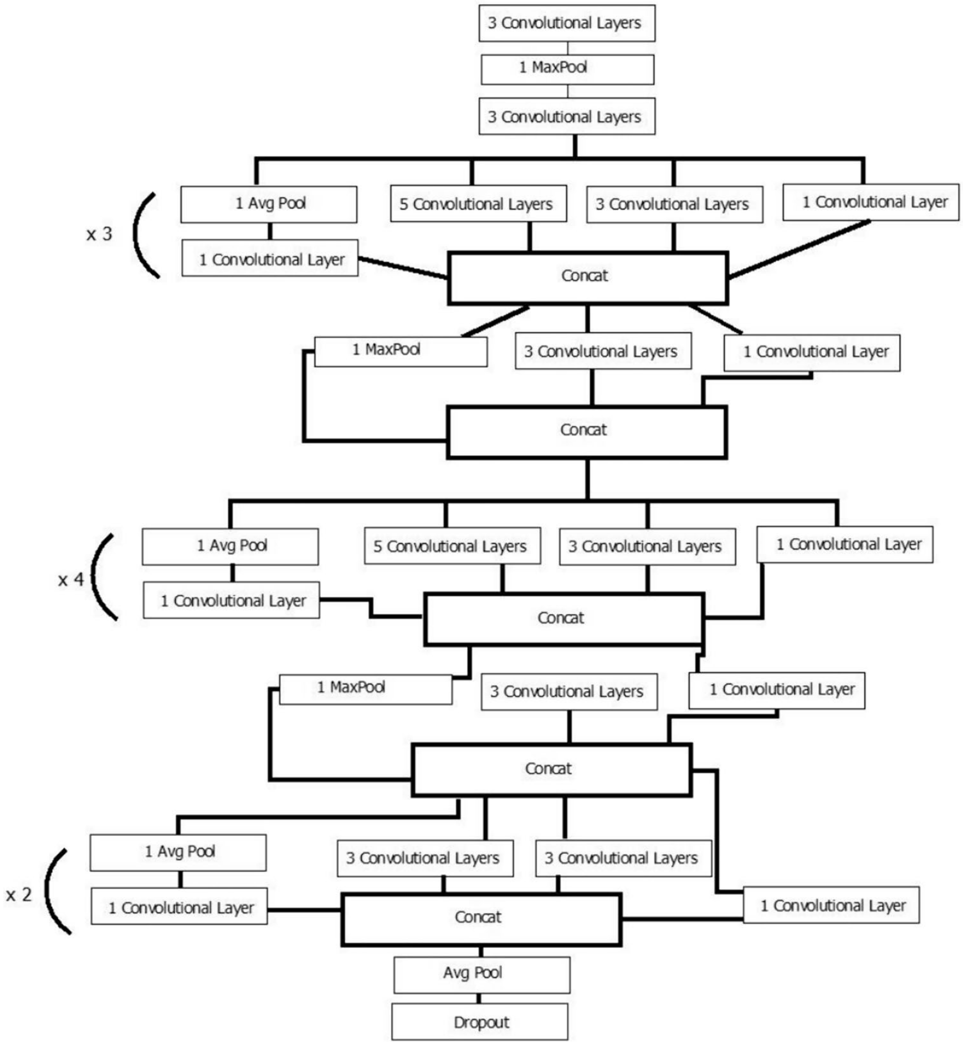


Fig. 1 InceptionV3 CNN architecture

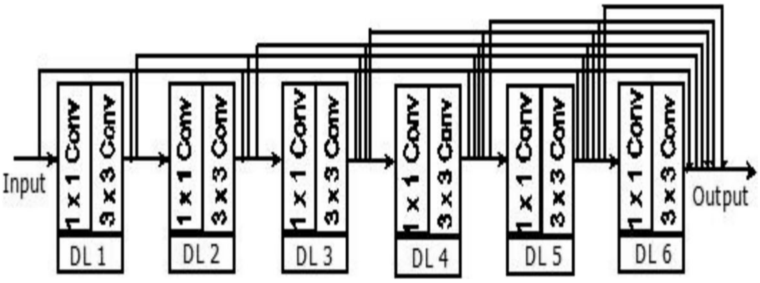


Fig. 2 Six dense layers on a dense block

serves as a fundamental foundational element in the network [39].

Equation 7 shows the number of network connections.

$$xl = H\left(\left[x_0, x_1, x_{2,} \ldots \ldots \ldots x_{l-1}\right]\right) \tag{7}$$

where [×0, X1…, xl1-1] are the input layers that combine the feature maps and generate the output xl.

Figure 3 shows the DenseNet121 CNN model, a typical DenseNet network implementation consisting of two main fragments: dense (DB) and transitional (TB) blocks. Each layer forms a thick alliance with one convolution layer of size 1 × 1 and three 3 × 3 sizes of convolution layers. In the spaces between the thick blocks, Transition blocks connect some of the blocks with a batch normalization layer, then a convolution layer of 1 × 1 dimension, and finally a 2 × 2 range average pooling layer. In this implementation, there are four dense blocks. Depending on the density of the block, there are six, twelve, twenty-four, or sixteen layers of thickness.

Feature Fusion

By combining multiple features or representations extracted from different sources or modalities, feature fusion creates a more comprehensive model suitable for various applications, including object recognition, image segmentation, and speech recognition. Fusion algorithms examine the input tensors of dimensions $m_{x,y,d} and m'_{x,y,d}$ for x, y and d-dimensions, respectively. In addition, each input tensor is combined element by element to produce another tensor with the exact dimensions as previous tensors represented by Eq. 8. Equation 9 represents the sum and the aspect taken for each input tensor, the highest value element, and then an identical tensor with the same shape is produced. If we consider averaging instead of maximisation, then the method known as the average method is represented by Eq. 10. The input tensors are stacked together using the concatenation method along the axis. That is represented by Eq. 11.

$$Y^{sum}_{x,y,d} = m_{x,y,d} + m'_{x,y,d} \tag{8}$$

$$Y^{max}_{x,y,d} = \max\left(m_{x,y,d}, m'_{x,y,d}\right) \tag{9}$$

$$Y^{avg}_{x,y,d} = avg\left(m_{x,y,d}, m'_{x,y,d}\right) \tag{10}$$

$$Y^{concat}_{x,y,2d} = m_{x,y,d}, Y^{concat}_{x,y,2d-1} = q_{x,y,d} \tag{11}$$

Figure 4 portrays the comprehensive procedure of the proposed work, encompassing three main stages: data prepraration, training, and evaluation.

Description of Dataset

Throughout this experiment, using Kaggle's Plant Village dataset [40], Authors classified 3076 images of healthy and diseased coffee leaves into four groups of healthy coffee leaves and three types of diseased leaves, including rust on coffee leaves, black rot, and brown spot. The numbers of images and samples of images are shown in Table 1 and

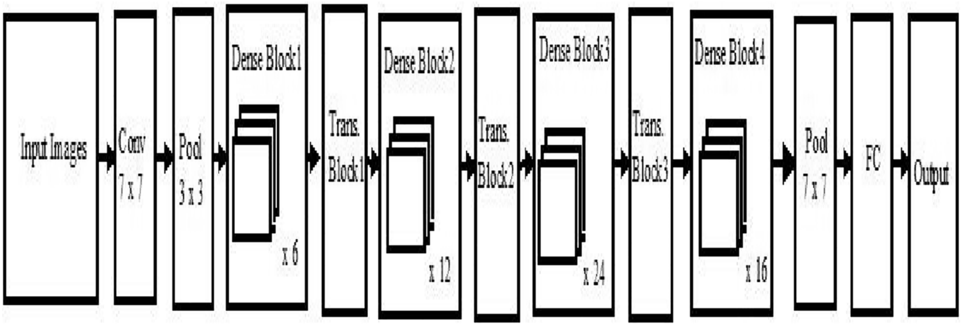


Fig. 3 DenseNet121 CNN architecture

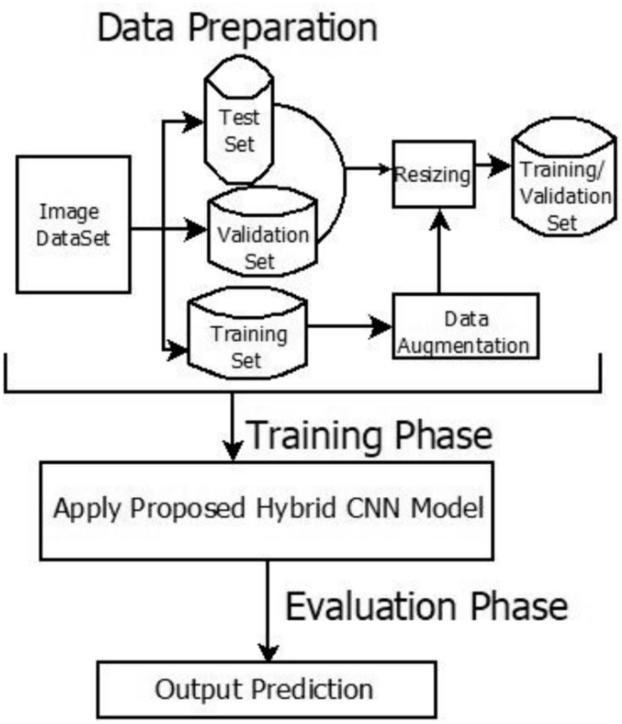


Fig. 4 Proposed procedural framework for a classification procedure

Table 1 The combined count of images used across all categories within the datasets

Category of leafs	Number of images
coffee leaf rust	408
black rot coffee leaves	952
brown eye spot coffee leaves	788
Healthy coffee leaves	928
Total	3076

Fig. 5, respectively. Sometimes little patches of pale yellow can be seen within the upper surface of coffee leaves, called coffee leaf rust [17]. The lower surface of the leaf experiences a gradual size expansion of these spots. They increase in size and vary from orange–yellow to orange–red. The eye-like appearance of coffee leaves is known as brown eye spot disease. Small chlorotic spots on leaves that grow to a 3/16-to-5/8-inch diameter turn brown. The dark brown or black appearance of infected coffee leaves is known as "black rot". Sclerotia cover these on the petioles and twigs. Using machine learning techniques, the experimental dataset was split into training and test sets to evaluate the model's performance on unseen data [41, 42]. Model training typically utilizes 80% of the information, and Testing reserves the remaining 20%. Further, hyper parameter tuning keeps

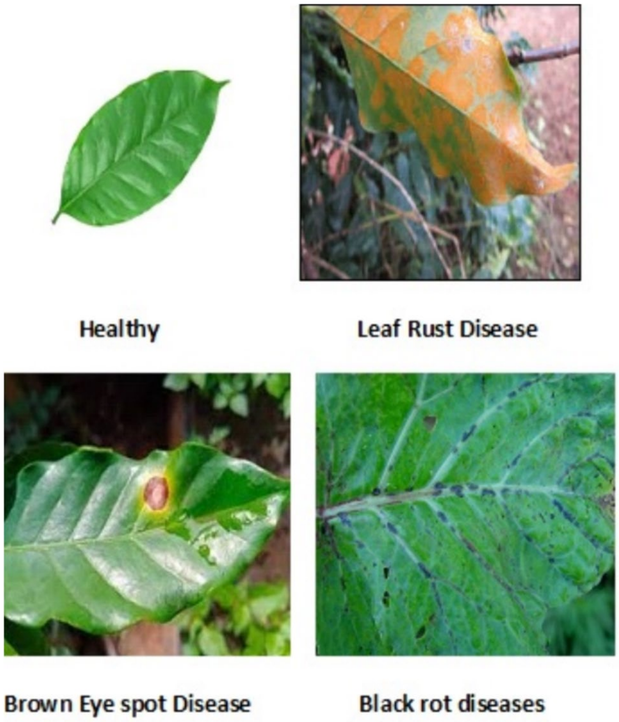


Fig. 5 The dataset includes examples representing each category: Healthy, coffee leaf rust, brown eye spot and black rot

Table 2 Data augmentation values to be used

Data augmentation	Values
Zoom percentage	30
Rotation angle	30°
Horizontal image flipping	Yes

Distinct from the training subset into the validation subset. The model is evaluated on data not observed during training, thanks to this process. To ensure well-training performance and prevent overfitting, a validation subset is employed. Table 2 displays the available augmented methods. Before further processing, the size standardizes to 244 × 244 pixels with a uniform standard.

Training Phase

To initialize the new proposed model, we utilized pre-trained weights from DenseNet121 and InceptionV3 CNNs, as depicted in Fig. 6. We amalgamated the classification layers with a fully connected layer comprising 1024 neurons and a median pooling layer. Additionally, three extra fully connected layers, with 256, 512, and 1024 neurons, respectively, replaced both branches. Final pruning was executed by applying a hierarchical

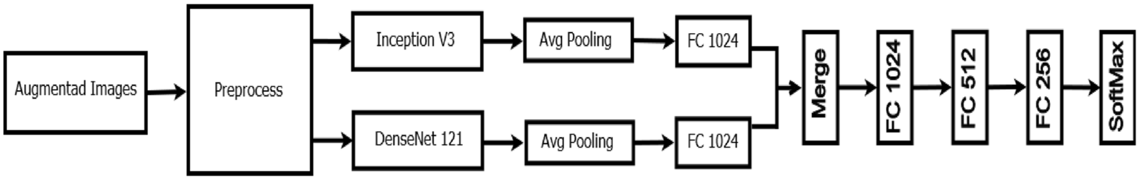


Fig. 6 Proposed Hybrid CNN architecture

cross-entropy loss function to the softmax layers of four neurons. Considering that DenseNet121 and InceptionV3 necessitate different pre-processing, we incorporated a pre-processing layer for feature extraction before training. This pre-processing layer facilitated feature extraction to accommodate the distinct pre-processing requirements of DenseNet121 and InceptionV3.The proposed CNN architecture, as illustrated in Fig. 6, encompasses two branches, each fulfilling its function for pre-processing, feature extraction, and a fully linked layer with average pooling. The outputs of the fully connected layers in each branch are amalgamated using concatenation. The concatenated output proceeds through several connected layers and culminates with a softmax classification layer.

Phase of Evaluation

The authors tested the model during the assessment stage to assess its effectiveness and ability to prevent the "overfit" problem. For testing and training, 80% and 20% of samples from the original dataset includes, respectively. During an evaluation, the image divides into four groups like Healthy, Coffee Leaf Rust, Brown Eye Spot and Black Rot. The proposed model used the Keras library. It provides a wide range of pre-defined layers, activation functions, loss functions, optimisers, and utilities to streamline the process of building and training models. This model runs on Intel i7 CPU, Equipped with 250 GB of Solid-state drive (SSD) and random-access memory (RAM) with up to 16 GB of storage capacity. Cross-entropy is often used for training models as a loss or objective function, particularly in classification tasks. For multi-class problems with K classes, the formula for categorical cross-entropy is:

$$CE = \sum [y * \log(p)] \tag{12}$$

where y is the true a one-hot vector for the probability distribution, and p represents the predicted probability distribution across all classes. The limitations of our proposed model include the possibility of heightened system complexity resulting from the integration of diverse models and datasets. Fine-tuning the entire structure might prove intricate and time-consuming. Additionally, if rigorous validation is required, there could be susceptibility to overfitting.

Results and Discussions

Initial training subset and constructed a hybrid CNN model using the augmented training set illustrated in Fig. 7, the researchers trained the hybrid CNN model. The training process begins with Learning at a rate of 0.01. The learning rate reduces by 0.1 after every five epochs, with no adjustment to the validation loss value for 50 iterations. Model training stops if there is no improvement for ten consecutive periods. Other CNN models, such as InceptionV3, DenseNet121, EfficientNetB0, and the suggested hybrid model, also undergo this training. In multi-class classification problems, frequently, classifiers use Positive and Negative True and False TP, TN, FP and FN, respectively, to evaluate their performance. We can construct a confusion matrix for each class using these concepts by employing the one-versus-all strategy that allows us to assess each class classifier's effectiveness independently. Several criteria, such as F1 scores, recall, accuracy, and precision, are computed to evaluate each class classifier's performance. These metrics offer valuable information about the classifier's effectiveness for individual classes. Accuracy evaluates the overall correctness of the classifier's predictions for a type. Precision, among all predicted positive cases, measures the proportion of accurately forecasted positive events. A recall calculates the rate of accurately predicted positive events compared to all positive samples. The F1 score offers a balanced evaluation of the classifier's performance. Analyzing these metrics for each class can thoroughly understand the classifier's

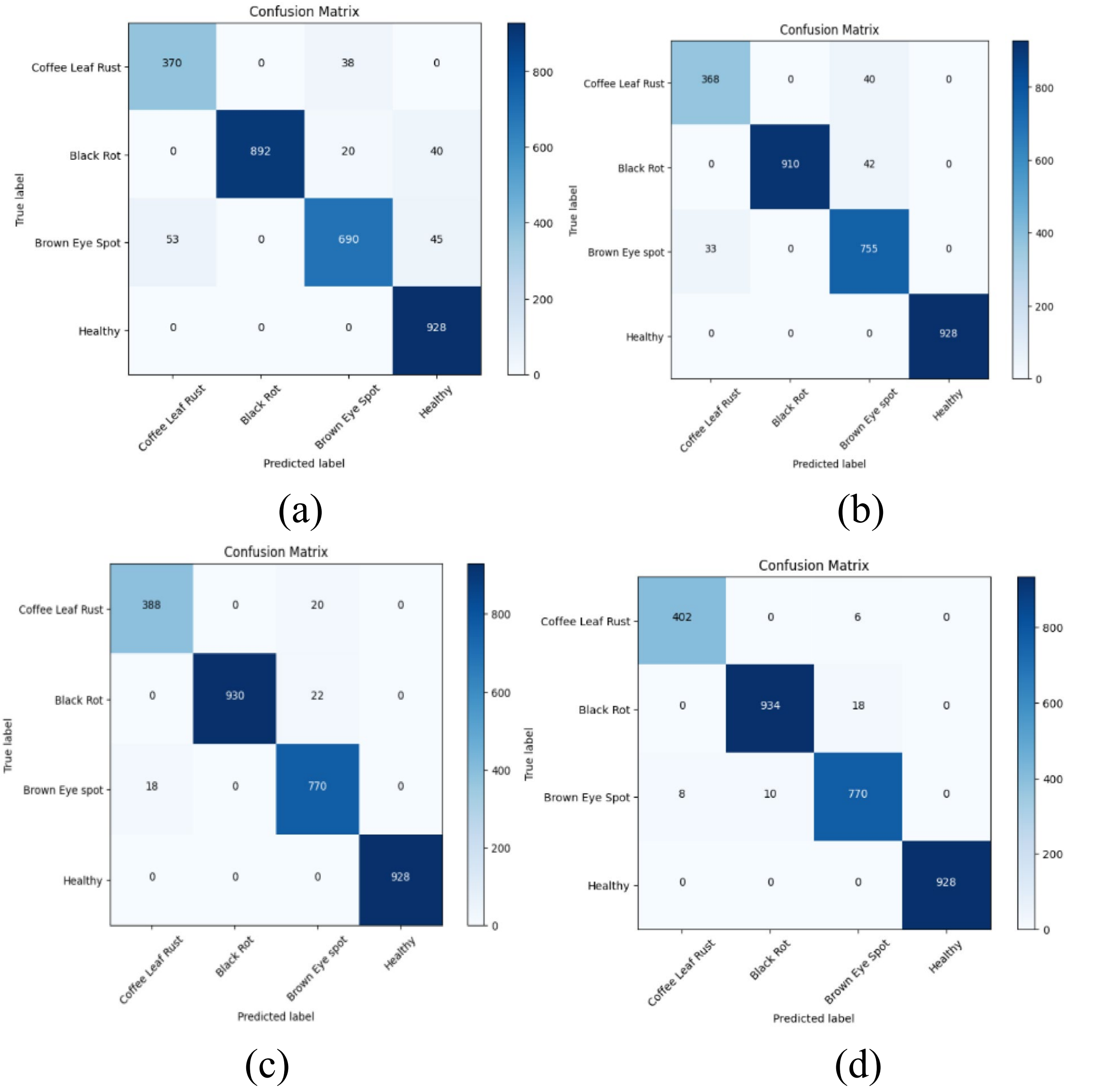


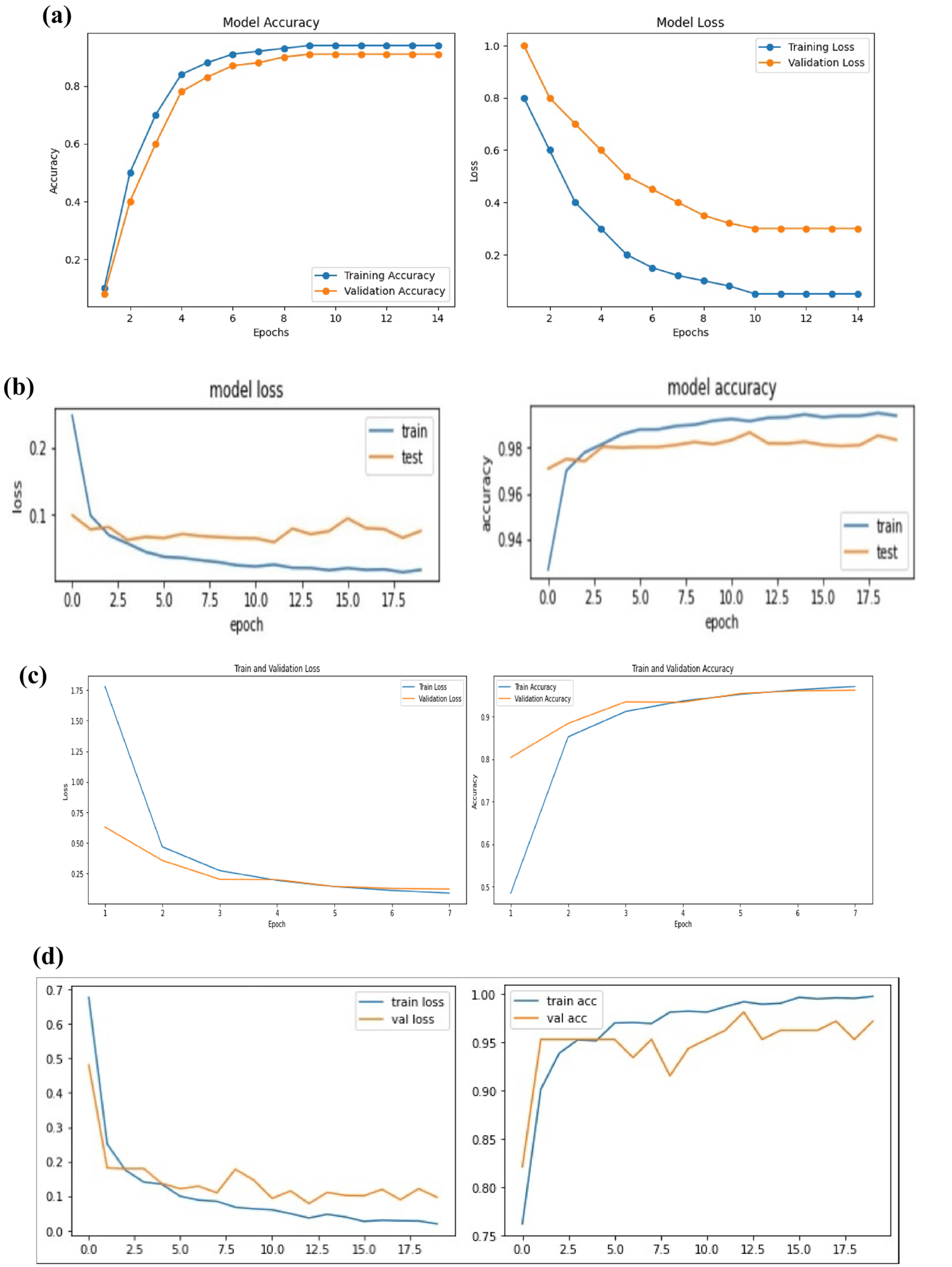
Fig. 7 The adapted models' confusion matrix for the experiment includes **a** InceptionV3 **b** DenseNet121, **c** EfficientNetB0 and **d** Proposed model

advantages and weaknesses in classifying multiple categories. This approach allows us to evaluate and compare the performance of individual class classifiers effectively.

Accuracy: Eq. 13 evaluates the ratio of accurate predictions to the total predictions generated by the model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

Precision: Eq. 14 shows that precision measures the ratio of accurate optimistic predictions to positive predictions.


(a)
Model Accuracy
Accuracy
0.2
0.4
0.6
0.8
Epochs
2
4
6
8
10
12
14
Training Accuracy
Validation Accuracy
Model Loss
Loss
0.2
0.4
0.6
0.8
1.0
Epochs
Training Loss
Validation Loss
(b)
model loss
loss
0.1
0.2
epoch
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
train
test
model accuracy
accuracy
0.94
0.96
0.98
epoch
train
test
(c)
Train and Validation Loss
Loss
0.25
0.50
0.75
1.00
1.25
1.50
1.75
Epoch
1
2
3
4
5
6
7
Train Loss
Validation Loss
Train and Validation Accuracy
Accuracy
0.5
0.6
0.7
0.8
0.9
Epoch
Train Accuracy
Validation Accuracy
(d)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
train loss
val loss
0.75
0.80
0.85
0.90
0.95
1.00
train acc
val acc

◂**Fig. 8** (**a**) Accuracy and the Loss of Validation and Training for InceptionV3. (**b**) Loss and the Accuracy of Validation and Training for DenseNet121. (**c**) Loss and the Accuracy of Validation and Training for EfficientNetB0. (**d**) Loss and the Accuracy of Validation and Training for Proposed model

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

Recall: Eq. 15 measures the ratio of accurate optimistic predictions from proper positive examples.

$$Recall = \frac{TP}{TP + FP} \tag{15}$$

F1 score: Eq. 16 shows the use of the F1 score when the misclassification of positive examples is as essential as negative examples.

$$F1score = \frac{(2 * Precision * Recall)}{Precision + Recall} \tag{16}$$

The computational accuracy of the trained models, such as InceptionV3, DenseNet 121, EfficientNet B0 and the proposed model, was assessed by evaluating their F1-score, precision and recall. Subsequently, we used the operating characteristics of the receiver (ROC) curve on the testing dataset. To analyse the predictions of the suggested model. The overall predictive accuracy of a model refers to its capacity to make accurate predictions of the outcome or label of a given dataset by using ROC measurement. The graph in Fig. 8 generates by comparing the recall value and the false positive rate in a (FPR) value at various threshold levels. The equation representing the FPR denotes Eq. 17.

$$FPR = \frac{FP}{FP + TN} \tag{17}$$

The primary purpose of the proposed work was to rapidly build and develop a deep learning model for which we used the Keras library. We conducted the experiment utilizing the Google Collaboratory (Colab) platform and an Intel processor equipped with 16 GB of RAM. This study closely explored the variance between the real and forecasted output with the help of cross-entropy (CE). Adam's algorithm improves on the difference obtained, minimising the cost function so that the model performs better. After the initial training is augmented, the suggested hybrid approach shown in Fig. 6 trains CNN. The learning rate for training initially takes as 0.02 for a total of 50 epochs. After that, the learning rate is reduced to 0.1 if there is no improvement after every four generations. This procedure uses for InceptionV3, DenseNet 21, EfficientNet B0 and the proposed hybrid model. Figure 8 illustrates the classification accuracies obtained on the same dataset by various models, namely EfficientNetB0, DenseNet121, InceptionV3, and the proposed hybrid model, which were recorded as 98%, 96.25%, 94%, and 99%, respectively. Table 3 exhibits the accuracy of the proposed hybrid model along with EfficientNet B0, DenseNet 121, and InceptionV3, whereas Tables 4, 5, 6, and 7 showcase the calculated recall, accuracy, and F1 scores of these models in the research. The result shows that the proposed hybrid model achieved the maximum F1 score for coffee leaf rust with the highest accuracy, outperforming InceptionV3, DenseNet 121 and EfficientNet B0. The confusion matrix displayed in Fig. 7 shows the classification of the suggested hybrid model, EfficientNet B0, DenseNet 121 and InceptionV3 into coffee leaf rust, black rot, brown eye spot and healthy leaves, where it is apparent that the proposed hybrid has a better model performance. Figure 9 displays the ROC curve of the proposed model.

Table 3 Accuracy of the experimental models

Model	Accuracy %
InceptionV3	94
EfficientNetB0	98
DenseNet121	96.25
Proposed Hybrid CNN Model	99

Conclusion and Outline of Future Work

The results conclude that improving the classification accuracy of coffee plant disease requires a combination of in-depth features that achieve by producing different CNNs and complex feature sets. This work concludes that the presented model performed well compared to other methods, with 99% classification accuracy. In this work, we used CNN with few parameters and combined feature sets to generate a robust model. The model performed very well with a large dataset. In the future, we will study more and more diseases so that more situations detect from the beginning stage. Not only this but also the use of other augmentation methods to boost the detail

Table 4 The models' accuracy, recall, and F1-score in InceptionV3

Class	Precision (%)	Recall (%)	F1-score (%)
Coffee Leaf Rust	87	91	89
Black Rot	100	94	97
Brown Eye spot	92	88	90
Healthy	92	100	100
		Accuracy	94%

Table 5 The models' accuracy, recall, and F1-score in EfficientNetB0

Class	Precision (%)	Recall (%)	F1-score (%)
Coffee Leaf Rust	95	96	95
Black Rot	98	100	99
Brown Eye spot	98	95	96
Healthy	100	100	100
		Accuracy	98%

Table 6 The models' accuracy, recall, and F1-score in DenseNet121

Class	Precision (%)	Recall (%)	F1-score (%)
Coffee Leaf Rust	91	92	92
Black Rot	96	100	98
Brown Eye spot	96	90	93
Healthy	100	100	100
		Accuracy	96.25%

Table 7 The models' accuracy, recall, and F1-score in the proposed model

Class	Precision (%)	Recall (%)	F1-score (%)
Coffee leaf rust	98	98	98
Black rot	98	99	99
Brown eye spot	98	97	97
Healthy	100	100	100
		Accuracy	99%

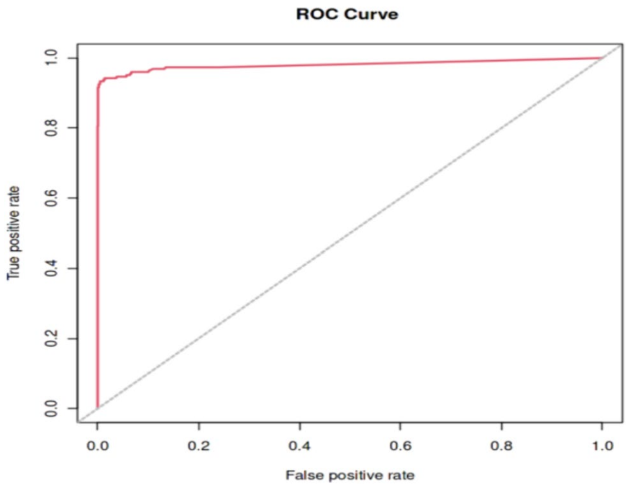


Fig. 9 ROC curve for the suggested model

of the image can make the model more useful. In future, we will also test other hybrid techniques and create new models by hybridising more than two models to get better results. In our hybrid model, we can extract additional insights by incorporating diverse datasets, including cucumber, pumpkin, grapes, and more. The application of our proposed model allows for precise predictions not only in plant-related contexts but also in the analysis of biological data, such as breast cancer, Parkinson's, and various other human diseases.

Author Contribution Equal contributions were made to this study by each author. The manuscript's published version was approved by all authors after they had read it.

Funding Not Applicable.

Data Availability For this work, 3076 images of various coffee leaves, i.e., Coffee Leaf Rust, Black Rot, Brown Eye spot, Healthy, have been taken from Plant Village's vast repository.

Declarations

Conflicts of Interest The authors declare that they have no competing interests.

Research Involving Human and /or Animals Not Applicable.

Informed Consent Not Applicable.

References

1. Esquivel P, Jimenez VM. Functional properties of coffee and coffee by-products. Food Res Int. 2012;46(2):488–95. https://doi.org/10.1016/j.foodres.2011.05.028.
2. Sousa LP, Silva MJ. JM Costa Mondego, Leaf-associated bacterial microbiota of coffee and its correlation with manganese and calcium levels on leaves. Genet Mol Biol. 2018;41:455–65. https://doi.org/10.1590/1678-4685-GMB-2017-0255.
3. GAIN, USDA. USDA GAIN: Ethiopia Grain and Feed Annual Report. GAIN Report Number: ET1401, (2014)
4. Degum H, Chewaka M. Review on coffee (Coffea arabica L.) wet processing more focus in Ethiopia. Acta Sci Agric. 2019;3(11):11–5. https://doi.org/10.31080/ASAG.2019.03.0676.
5. Belan LL, de Jesus WC, de Souza AF, Zambolim L, Filho JC, Barbosa DHSG, Moraes WB. Management of coffee leaf rust in Coffea canephora based on disease monitoring reduces fungicide use and management cost. Eur J Plant Pathol. 2020;156:683–94. https://doi.org/10.1007/s10658-019-01917-6.
6. Elhassouny A, Smarandache F (2019) Smart mobile application to recognize tomato leaf diseases using convolutional neural networks. Collected apers, pp 431
7. Manso GL, Knide H, Krohling RA, Ventura JA (2019) A smartphone application to detection and classification of coffee leaf miner and coffee leaf rust. arXiv preprint arXiv:1904.00742 (2019). https://doi.org/10.48550/arXiv.1904.00742

8. Toseef M, Khan MJ. An intelligent mobile application for diagnosis of crop diseases in Pakistan using fuzzy inference system. Comput Electron Agric. 2018;153:1–11. https://doi.org/10.1016/j.compag.2018.07.034.
9. Liebig T, Ribeyre F, Läderach P, Poehling HM, Asten PV, Avelino J. Interactive effects of altitude, microclimate and shading system on coffee leaf rust. J Plant Interact. 2019;14(1):407–15. https://doi.org/10.1080/17429145.2019.1643934.
10. de Resende MLV, Pozza EA, Reichel T, Botelho DMS. Strategies for coffee leaf rust management in organic crop systems. Agronomy. 2021;11(9):1865. https://doi.org/10.3390/agronomy11091865.
11. Ehrenbergerová L, Kučera A, Cienciala E, Trochta J, Volařík D. Identifying key factors affecting coffee leaf rust incidence in agroforestry plantations in Peru. Agrofor Syst. 2018;92:1551–65. https://doi.org/10.1007/s10457-017-0101-x.
12. Merle I, Pico J, Granados E, Boudrot A. Unraveling the complexity of coffee leaf rust behavior and development in different Coffea arabica agroecosystems. Phytopathology. 2020;110(2):418–27. https://doi.org/10.1094/PHYTO-03-19-0094-R.
13. Waller JM. Coffee rust—epidemiology and control. Crop Prot. 1982;1(4):385–404. https://doi.org/10.1016/0261-2194(82)90022-9.
14. Sylvain PG. Ethiopian coffee—its significance to world coffee problems. Econ Bot. 1958;12:111–39. https://doi.org/10.1007/BF02862767.
15. van der Graaff NA (1981) Selection of arabica coffee types resistant to coffee berry disease in Ethiopia. Wageningen University and Research
16. McDonald J. A preliminary account of a disease of green Coffee berries in Kenya Colony. Trans Br Mycol Soc. 1926;11:1–2.
17. Waller JM, Bigger M, Hillocks RJ. Coffee pests, diseases and their management. CABI, 2007.
18. Rayner RW. Coffee berry disease—a survey of investigations carried out up to 1950, plant pathologist. East Afr Agric J. 1952;17:130–58. https://doi.org/10.1080/03670074.1952.11664802.
19. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521:436–44. https://doi.org/10.1038/nature14539.
20. Fuentes A, Yoon S, Kim SC, Park DS. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors. 2017;17(9):2022. https://doi.org/10.3390/s17092022.
21. Ferrão RG, Fonseca AFA, Ferrão MAG, De Muner LH (2017) Café conilon. Instituto Capixaba de Pesquisa Assistência Técnica e Extensão Rural–INCAPER
22. Tassis LM, de Souza JET, Krohling RA. A deep learning approach combining instance and semantic segmentation to identify diseases and pests of coffee leaves from in-field images. Comput Electron Agric. 2021;186: 106191. https://doi.org/10.1016/j.compag.2021.106191.
23. Sorte LXB, Ferraz CT, Fambrini F, Roseli DRG, José HS. Coffee leaf disease recognition based on deep learning and texture attributes. Proc Comput Sci. 2019;159:135–44. https://doi.org/10.1016/j.procs.2019.09.168.
24. Rivillas CA, Serna CA, Gaitan AL (2011) La roya del cafeto en Colombia: Impacto manejo y costos del control
25. Li D, Dong Y. Deep learning: methods and applications, foundations and trends®. Signal Process. 2014;7(3–4):197–387. https://doi.org/10.1561/2000000039.
26. Amin H, Darwish A, Hassanien AE, Soliman M. End-to-end deep learning model for corn leaf disease classification. IEEE Access. 2022;10:31103–15. https://doi.org/10.1109/ACCESS.2022.3159678.
27. Singh MK, Kumar A. Cucumber leaf disease detection and classification using a deep convolutional neural network. J Inf Technol Manag. 2023;15(2):94–110. https://doi.org/10.22059/jitm.2023.95248.
28. Agarwal M, Bohat VK, Ansari MD, Sinha A, Gupta SK, Garg D. A onvolution neural network based approach to detect the disease in corn crop. In: 2019 IEEE 9th international conference on advanced computing (IACC), pp. 176–181. IEEE, 2019. https://doi.org/10.1109/IACC48062.2019.8971602
29. Eunice J, Popescu DE, Chowdary MK, Hemanth J. Deep learning-based leaf disease detection in crops using images for agricultural applications. Agronomy. 2022;12(10):2395. https://doi.org/10.3390/agronomy12102395.
30. Zhao Y, Liu L, Xie C, Wang R, Wang F, Bu Y. An effective automatic system deployed in agricultural internet of things using multi-context fusion network towards crop disease recognition in the wild. Appl Soft Comput. 2020;89: 106128. https://doi.org/10.1016/j.asoc.2020.106128.
31. Chakravarthy SS, Bharanidharan N, Rajaguru H. Deep learning-based metaheuristic weighted K-nearest neighbor algorithm for the severity classification of breast cancer. IRBM. 2023;44(3): 100749. https://doi.org/10.1016/j.irbm.2022.100749.
32. Lu SY, Wang SH, Zhang YD. BCDNet: an optimized deep network for ultrasound breast cancer detection. IRBM. 2023;44(4): 100774. https://doi.org/10.1016/j.irbm.2023.100774.
33. Liu T, Huang J, Liao T, Pu R, Liu S, Peng Y. A hybrid deep learning model for predicting molecular subtypes of human breast cancer using multimodal data. Irbm. 2022;43(1):62–74. https://doi.org/10.1016/j.irbm.2020.12.002.
34. Dhanaseelan FR, Sutha MJ. Detection of breast cancer based on fuzzy frequent item sets mining. Irbm. 2021;42(3):198–206. https://doi.org/10.1016/j.irbm.2020.05.002.
35. Akben SB. Determination of the blood, hormone and obesity value ranges that indicate the breast cancer, using data mining based expert system. Irbm. 2019;40(6):355–60. https://doi.org/10.1016/j.irbm.2019.05.007.
36. Rahman MM, Ghasemi Y, Suley E, Zhou Y, Wang S, Rogers J. Machine learning based computer aided diagnosis of breast cancer utilizing anthropometric and clinical features. Irbm. 2021;42(4):215–26. https://doi.org/10.1016/j.irbm.2020.05.005.
37. Ji M, Zhang L, Wu Q. Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks. Inf Process Agric. 2020;7(3):418–26. https://doi.org/10.1016/j.inpa.2019.10.003.
38. Tan M, Le Q (2019) Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning, pp. 6105–6114. PMLR, 2019. https://proceedings.mlr.press/v97/tan19a.html.
39. Iandola F, Moskewicz M, Karayev S, Girshick R, Darrell T, Keutzer K. Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869 (2014). https://doi.org/10.48550/arXiv.1404.1869
40. Esgario JGM, Krohling RA, Ventura JA. Deep learning for classification and severity estimation of coffee leaf biotic stress. Comput Electron Agric. 2020;169: 105162. https://doi.org/10.1016/j.compag.2019.105162.
41. Khan T, Karan S, Mohd S, Khaleel A, Savita KS, Ahmadian A, Salahshour S, Conti M. An efficient trust-based decision-making approach for WSNs: machine learning oriented approach. Comput Commun. 2023;209:217–29.
42. Khan T, Singh K, Ahmad K, Ahmad KAB. A secure and dependable trust assessment (SDTS) scheme for industrial communication networks. Sci Rep. 2023;13(1):1910.

43. Ahmed M, Afreen N, Ahmed M, Sameer M, Ahamed J. An inception V3 approach for malware classification using machine learning and transfer learning. Int J Intell Netw. 2023;4:11–8. https://doi.org/10.1016/j.ijin.2022.11.005.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.