Agrociencia

FUTURE-PROOF COFFEE PLANT DISEASE DETECTION BASED ON COUNTER-FACTUAL RECOMMENDATION WITH A HYBRID VISION TRANSFORMER AND CONVOLUTIONAL NEURAL NETWORK MODEL

Karthik **Selvaraj**[1*], Raveena **Selvanarayanan**[2], Sam Kumar **Gopalsamy Venkatesan**[3], Surendran **Rajendran**[4*]

[1]Muthayammal Engineering College (Autonomous). Department of Computer Science and Business Systems. Rasipuram 637408, Tamil Nadu, India.
[2]Panimalar Engineering College, Department of Computer Science and Business Systems, Chennai 600123, Tamil Nadu, India.
[3]Koneru Lakshmaiah Education Foundation. Department of Computer Science and Engineering, Vaddeswaram 522302, Andhra Pradesh, India.
[4]Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences. Department of Computer Science and Engineering. Chennai 602117, Tamil Nadu, India.
* Author for correspondence: surendran.phd.it@gmail.com

Citation: Selvaraj K, Selvanarayanan R, Gopalsamy Venkatesan SK, Rajendran S. 2025. Future-proof coffee plant disease detection based on counter-factual recommendation with a hybrid vision transformer and convolutional neural network model.
Agrociencia. https://doi.org/ 10.47163/agrociencia.v59i4.3385

Editor in Chief:
Dr. Fernando C. Gómez Merino

Received: December 16, 2024.
Approved: May 30, 2025.
Published in Agrociencia: June 06, 2025.

ABSTRACT

Coffee plantations are vulnerable to several diseases that harm roots, leaves, and cherries, jeopardizing crop productivity and farmer livelihoods. Small-scale farmers lack access to precise and accessible technologies for diagnosing and controlling these diseases. Traditional machine learning methodologies are restricted to single-disease classification and lack the intricacies of multi-disease contexts. In this work, the proposed model has a unique hybrid model that integrates vision transformer (ViT) and convolutional neural network (CNN) architectures for the identification and early detection of several coffee plant diseases. The ViT module identifies global associations in plant images, while the CNN extracts intricate local characteristics, facilitating thorough disease diagnosis. Furthermore, the counterfactual recommendation system models the impacts of several treatments and preventative strategies on the original images, offering practical insights. Our model attains an accuracy of 0.9881 % on a dataset of 1056 images, surpassing current methodologies. The suggested solution is included in the Affogato app, enabling farmers to make educated, customized choices about disease control. This method not only improves disease detection but also promotes sustainable coffee-growing techniques, enhancing crop production and farmer livelihoods.

Keywords: *Colletotrichum kahawae, Hemilieia vastatrix, Mycosphaerella coffeicola*, hybrid vision transformer, convolutional neural network, counter factual recommendation.

INTRODUCTION

Coffee is one of the most widely consumed beverages worldwide and plays a vital role in global agriculture and economies. However, coffee plants are highly susceptible to

various diseases that, if not promptly identified and managed, can lead to significant yield losses and reduced coffee quality. Coffee plant diseases pose a major threat to global coffee production, with leaf rust, root rot, and berry infections collectively contributing to over 30 % yield losses worldwide. These impacts are particularly severe for small-scale farmers who lack access to advanced diagnostic tools. Rapid detection and classification of infections are essential for mitigating economic losses and ensuring sustainable coffee production.

Several fungal, bacterial, and viral infections threaten coffee plants at various growth stages. coffee berry disease, caused by *Colletotrichum kahawae*, primarily affects *Coffea arabica* berries, leading to black, sunken lesions that spread rapidly, causing fruit decay and substantial economic losses. Similarly, brown eye spot disease (*Mycosphaerella coffeicola*) manifests as circular brown lesions with yellow halos on leaves and brown spots with fungal spore-containing gray cores in berries, leading to premature leaf abscission and reduced productivity. Leaf rust disease (*Hemileia vastatrix*) presents as yellow, greasy patches on the upper leaf surface, which progress into powdery orange pustules on the underside, ultimately causing defoliation and decreased coffee yield. Other severe infections, such as coffee wilt disease (*Fusarium xylarioides*) and coffee bark disease (*Fusarium stilboides*), disrupt plant vascular functions, leading to withering and mortality. Soilborne pathogens like *Pythium* and *Phytophthora* contribute to damping-off and root rot, resulting in significant seedling losses.

The sustainability of coffee production is highly dependent on soil conditions. Ensuring optimal soil quality is essential for maintaining consistent crop yields, as different coffee varieties require specific soil compositions. Effective soil management, particularly through fertilization techniques, plays a critical role in maximizing soil fertility and supporting long-term productivity. Organic fertilizers enhance soil fertility and nutrient retention while minimizing environmental impacts (Abdulsahib *et al*., 2025). Leguminous plants contribute to soil improvement by fixing atmospheric nitrogen, whereas biochar enhances soil structure and overall fertility. Although synthetic fertilizers supply concentrated nutrients that accelerate plant growth, their excessive use can negatively affect soil health. Bio-fertilizers, comprising beneficial microorganisms such as *Rhizobium* spp. and mycorrhizal fungi, represent a sustainable alternative. These agents improve nutrient uptake and positively influence the composition of soil microbiota (Dias *et al*., 2025). Furthermore, the integration of soil sensors with artificial intelligence (AI) systems enables real-time and accurate monitoring of key soil parameters, including pH, moisture, temperature, and nutrient levels, thereby supporting optimized irrigation and fertilization practices.

Emerging methods for crop analysis and forecasting have the potential to significantly enhance agricultural productivity. These approaches assist farmers in selecting crop varieties that are best suited to their specific climatic and soil conditions. In particular, machine learning techniques enable the automated identification of suitable crops as well as the detection of pests and diseases, thereby supporting farmers in maximizing yields while maintaining soil fertility and nutrient balance. In this study, seven

distinct machine learning algorithms were used for crop selection and yield estimation (Chaudhari *et al*., 2025). The recommended approach integrates soil composition and climate data to accurately predict the optimal crops for a given location. This form of crop recommendation holds promise for improving yield, sustainability, and profitability across a wide range of agricultural contexts. Through a comprehensive analysis of a large historical dataset and rigorous training and evaluation of multiple machine learning models, this study achieved a classification accuracy of 99.54 %, which represents the highest reported to date.

Advancements in artificial intelligence and imaging technologies have led to innovative approaches in disease detection and management. Conventional disease detection methods rely on expert knowledge and manual assessment, making them time-consuming and subjective. To overcome these limitations, machine learning and image processing techniques have been increasingly explored for automated and efficient disease classification (Signo *et al*., 2024). Early studies have demonstrated the effectiveness of machine learning-based models in classifying coffee leaf diseases. For instance, Coffee-Net, a deep learning approach, has achieved high precision in disease identification under controlled conditions. However, existing models primarily focus on single-disease classification, failing to account for multi-disease interactions. Additionally, environmental factors such as lighting variability, camera quality, and background noise significantly affect the accuracy and robustness of these models.

Contemporary machine learning architectures aim to enhance disease classification by integrating both global and local feature extraction methodologies. Using global context and local feature analysis helps to understand both the big picture and the small details of lesions, which greatly improves the ability to identify disease spots (Alharbi *et al*., 2023). Furthermore, the inclusion of environmental variables in disease prediction models is crucial for the development of effective management strategies. Advanced forecasting models that utilize climatic data and disease progression trends can support outbreak prediction and optimize resource allocation for coffee producers (Anand *et al*., 2024).

Beyond disease detection, machine learning has also been applied in broader domains related to coffee quality assessment. For example, coffee sample adulteration has been detected using an electronic nose (E-Nose) system equipped with volatile organic compound (VOC) sensors. In this context, the density-based spatial clustering of applications with noise (DBSCAN) algorithm has demonstrated efficacy in distinguishing between pure and adulterated coffee blends. Additionally, machine learning techniques have been explored in various agricultural and energy-related applications, such as wind power forecasting through observer-controller-based frameworks incorporating a modified flower pollination algorithm (M-FPA).

Despite recent advances, machine learning methods for coffee disease diagnosis continue to face challenges in real-world applications. The limitations of single-disease classification models, coupled with environmental constraints, underscore the need for more robust and comprehensive diagnostic frameworks (Duhan *et al*., 2025). To address

these shortcomings, this study proposes a hybrid vision transformer-convolutional neural network (ViT-CNN) model for the detection of coffee plant diseases. The model integrates global and local features to enhance disease classification performance under diverse field conditions. Additionally, a counterfactual recommendation system is incorporated to support personalized management strategies for coffee farmers, thereby enabling earlier diagnosis and more effective responses. The Affogato mobile application delivers real-time diagnostic insights to farmers, ultimately aiming to increase crop yields and promote sustainable coffee cultivation.

MATERIALS AND METHODS

A substantial methodological groundwork was used to create the hybrid ViT-CNN model for thorough coffee plant health monitoring. Multi-angle images of leaves, cherries, and flowers were taken utilizing unmanned aerial vehicles (UAVs), plantation-fixed cameras, and high-resolution sensors. To increase model generalization, preprocessing procedures included augmentation, contrast enhancement, and noise reduction. CNN for localized spatial feature learning and ViT for global feature extraction were combined in the hybrid architecture.

A large annotated dataset was used to train the model, including data-driven hyperparameter tweaking, batch normalization, and adaptive learning rates. Accuracy, precision, recall, F1-score, and area under the curve of the receiver operating characteristic (AUC-ROC) were used to assess performance, and benchmark deep learning models, including ResNet, DenseNet, and Inception were compared. By effectively addressing the shortcomings of traditional methods and improving early disease detection for optimal coffee yield management, the results displayed through statistical analyses, confusion matrices, and visual segmentation maps showed superior disease classification accuracy and predictive reliability.

Image segmentation, object detection, and classification

Images of coffee berries, cherries, leaves, flowers, roots, and stems from public websites and captured directly at coffee plantations served as the basis for the study (Shafik *et al*., 2025). Real-time monitoring of coffee plantations using IoT sensors was conducted to gather data on soil and fertilizer use. The dataset included 640 photos of symptomatic coffee cherries, 180 leaves exhibiting damping-off symptoms, 250 occurrences of root rot, and 1840 soil monitoring images at a resolution of 256 × 256.

Images were preprocessed using Python libraries for computer vision, augmented using normalization, augmentation, and noise reduction approaches to optimize input for deep learning models. The YOLOv8 model was used for object identification, enabling rapid and precise training for the investigation of coffee plant diseases. Comparisons were conducted using the Efficient-Det, SSD, and R-CNN models. The result comprises bounding boxes labeled as “Leaf” or “Coffee Cherry.” For image classification, ResNet101 functioned as the principal pre-trained model, exhibiting

enhanced performance in complexity and accuracy relative to VGG-Net, InceptionV3, InceptionV4, and Mobile-Net (Chinnasamy *et al*., 2023). This phase classified photos using designations such as "Healthy" or "Diseased."

Image segmentation was used to identify distinct regions within images based on color, texture, spatial data, and object boundaries. The objects of interest encompassed "leaves," "buds," "secondary stems," and "primary branches" of coffee plants. DeepLabV3+ was chosen for segmentation tasks, exhibiting enhanced performance on intricate datasets relative to U-Net, R-CNN, and Mask R-CNN models. The segmentation procedure was executed utilizing open-source PyTorch libraries for model development and training (Essa and Murshid, 2023).

A three-stage deep learning pipeline was developed for coffee plant disease detection (Rehman *et al*., 2025): 1): object detection to locate diseased areas within images; 2): segmentation of detected areas to differentiate healthy and unhealthy portions of leaves; and 3): classification of segmented regions to confirm the presence or absence of disease (Figure 1).

Coffee plant disease analysis using hybrid ViT-CNN

Vision transformers (ViTs) and convolutional neural networks (CNNs) were effectively integrated to enhance performance (Isinkaye and Erute, 2022). C CNNs extract local

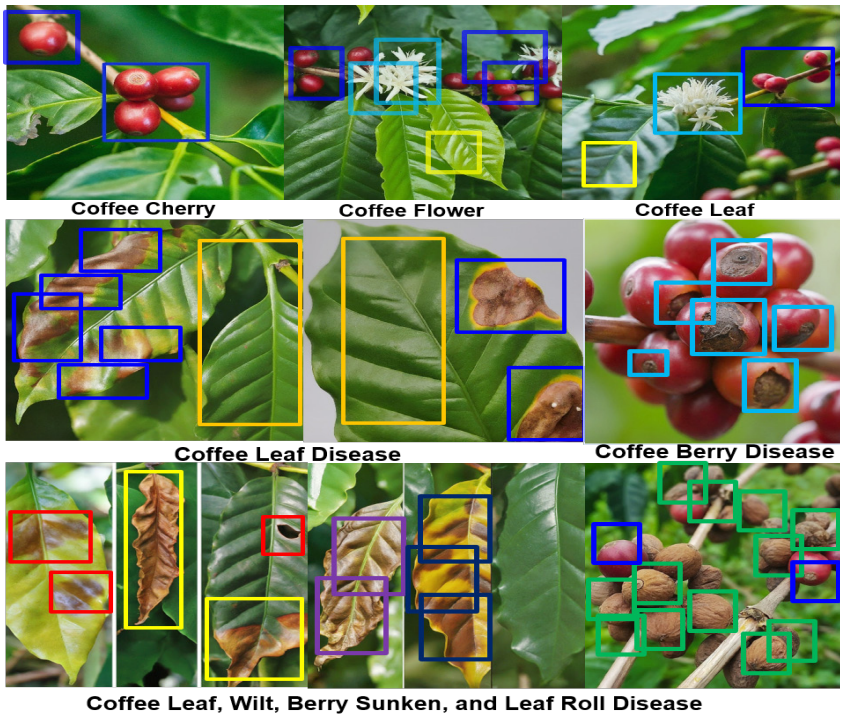

Figure 1. Object detection and instance segmentation demonstration on collected coffee plant images using YOLOv8. Bounding boxes (colored rectangles) indicate detected objected such as healthy coffee cherry, diseased leaves, and flowers.

features, while ViTs learn global dependencies, which have the potential to improve interpretability. The classified image dataset is split into rectangular patches. These patches are flattened into 1D vectors (height *H*, width *W*, and channel *C*) (Equation 1). A linear projection is applied to these vectors, transforming them into a fixed-size representation (Equation 2) where W_e is the embedding weight matrix and b_e is the embedding bias vector. Both W_e and b_e are learned during training. Linear projection consists of two main steps: first, flattening a vector, and second, transforming it into a fixed-size representation. This operation is performed twice in the original text and is thus reiterated here for completeness. Furthermore, the function *pow* increases exponentially with the index *i*, using a base of 10 000. These exponentially increasing values are used to generate different frequencies for the sine and cosine functions used in positional encodings.

$$flattened_{patch} = reshape(patch, [1, H * W * C]) \tag{1}$$

$$embedding = W_e * flattened_{patch} + b_e \tag{2}$$

$$PE(pos, i) = \sin\left(\frac{pos}{pow\left(10\,000, 2 * \frac{i}{d_{model}}\right)}\right) if\ i\ is\ even \tag{3}$$

$$PE(pos, i) = cos\left(\frac{pos}{pow\left(10\,000, \frac{2 * i + 1}{d_{model}}\right)}\right) if\ i\ is\ odd \tag{4}$$

$$encoded_{patch} = embedding + PE(pos) \tag{5}$$

Each image patch is effectively compressed into a lower-dimensional space through this method. To retain spatial information, positional encoding is incorporated into the embedding, thereby enabling the model to capture the relative position of each patch within the original image. Common approaches for positional encoding use sine and cosine functions (Equations 3 and 4), which define the positional encoding (PE) for a given position pos and embedding dimension *i*. The model dimension corresponds to the size of the output embedding vector. By utilizing sine and cosine functions at different frequencies across each dimension, this method generates a position-specific encoding with values ranging from -1 to 1. Consequently, for each component of the embedding vector, a position-specific encoding is established (Equation 5).

Multi-Head Self-Attention (MHA)

The interrelationships among various components within an image prompt the model to generate multiple "heads," each functioning as an autonomous attention mechanism that analyzes the image from a distinct perspective. Through linear projections, each head is transformed into three separate vectors: the Query (*Q*), the Key (*K*), and the Value (*V*) (Jiang *et al*., 2025). The Query vector is responsible for locating the portion of the image that potentially contains the disease. The Key vector encodes the salient features of the disease, such as the color, shape, and texture of lesions on the leaf. Meanwhile, the Value vector stores supplementary information relevant to the disease, including its name, its impact on the plant, associated soil conditions, required fertilizers, and possible treatment options:

$$Q = WQ * X + bQ \tag{6}$$

$$K = WK * X + bK \tag{7}$$

$$\mathrm{V} = \mathrm{WV} * \mathrm{X} + \mathrm{bV} \tag{8}$$

where *X* is a flattened image patch (d-dimensional vector); *WQ*, *WK*, and *WV* are weight matrices for Query (Equation 6), Key (Equation 7), and Value (Equation 8) projections; and *bQ*, *bK*, and *bV* are bias vectors for Query, Key, and Value projections. Attention scores assign a score to each pair of patches, indicating how relevant one patch (Value) is to the query of another. This score is calculated by taking the dot product of the Query and Key vectors for each pair (Jung *et al*., 2023). The compute inner product function calculates the inner product of a specific patch's query vector (*Q*) and the key vector (*K*) of all patches in the image. This operation indicates how close the current patch is to other areas of the image (Equation 9):

$$A_{ij} = Q_i * K_j \tag{9}$$

where *i* refers to the current patch and *j* iterates over all patches. The SoftMax function is applied to the inner products to normalize the attention scores and ensure they total to one. This function transforms them into weights that reflect the relative significance of each patch, depending on its resemblance to the current patch (Equation 10).

$$Attention(Q_i, K) = Softmax(A_i) \tag{10}$$

Attention scores are computed using weighted values, whereby the model produces a weighted sum of the Value (*V*) vectors corresponding to each patch. Patches assigned higher attention scores exert a greater influence on the final output (Lee and Rianto, 2024). Specifically, these scores are obtained by multiplying the attention weights, derived from the SoftMax function with the respective Value vectors across all patches.

This mechanism serves to emphasize the contributions of the most relevant patches, as dictated by their attention scores (Equation 11).

$$Weighted_{Value_i} = Attention(Q_i, K) * V \quad (11)$$

The context vector that includes information from relevant parts of the image based on their similarity to the current patch (Equation 12).

$$Z_i = \Sigma\left(Weighted_{Value_i}\right) \quad (12)$$

Every head generates concatenated outputs to gather data from many angles. Applying a last linear projection helps to alter the result overall.

Feed-forward network with convolutional layers

Convolutional neural networks (CNNs) layers use image processing techniques to identify features such as edges, textures, and color patterns in the input image (coffee plant leaf, coffee berry, and flower) (Wang *et al*., 2025). By layering many convolutional layers, the network progressively enhances its comprehension of complex features that may be used for illness identification.

A feed-forward network (FFN) is a class of neural networks composed of multiple perceptrons, each of which uses a nonlinear activation function, specifically the Gaussian error linear unit (GELU). In the context of multiclass classification tasks, the primary loss function utilized is cross-entropy loss, which is widely used for predicting multiple disease categories, such as coffee leaf rust and berry disease. For the convolutional neural network (CNN) component within the hybrid model framework, stochastic gradient descent (SGD) with momentum is applied. This optimization strategy enhances convergence by promoting a more stable and directed trajectory throughout training iterations, thereby reducing oscillations.

These functions allow the network to learn about non-linear connections between the retrieved characteristics, which is necessary for distinguishing between healthy and diseased leaves with minor differences, as shown in the following algorithm:

```
def create_model(img_shape): # Define model architecture
    model = models.Sequential()
    model.add(layers.Conv2D(32, (3, 3), activation=" relu", input_shape=img_shape))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(64, (3, 3), activation="relu"))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Flatten())
    model.add(layers.Dense(128, activation="relu"))
    model.add(layers.Dropout(0.2)) # Optional for regularization
```

```
    model.add(layers.Dense(3, activation="softmax")) # Multi-class for healthy/
    diseased/other
    model.compile(loss="categorical_crossentropy", optimizer="adam",
    metrics=["accuracy"])
    return model
train_data, train_labels, test_data, test_labels = load_data() # Load preprocessed data
(replace with your data loading logic)
img_shape = train_data.shape[1:] # Define image shape based on your data
    model = create_model(img_shape) # Create and train the model
    model.fit(train_data, train_labels, epochs=10, validation_data=(test_data, test_
    labels))
loss, accuracy = model.evaluate(test_data, test_labels) # Evaluate the model on
unseen data
print ("Test Accuracy:", accuracy)
new_image = load_new_image() # Use the model for prediction on a new image
(replace with your image loading logic)
prediction = model.predict(np.expand_dims(new_image, axis=0))
predicted_class = np.argmax(prediction) # Get class with highest probability
```

The network incorporates a ViT module that emphasizes local features. ViT excels at capturing long-distance connections throughout the image (Shafik *et al*., 2025). This is especially useful for detecting diseases in coffee plants, as disease symptoms can be found throughout the leaf rather than just one area. The convolutional layers, along with feed-forward networks (FFNs), act as feature extractors, identifying patterns and textures in the coffee leaf image. The ViT module then examines the interrelations of the extracted characteristics over the whole leaf. Combining these functionalities allows the model to efficiently acquire the complex visual cues associated with various coffee plant diseases.

Batch normalization

Batch normalization layers can be positioned after every convolutional layer and fully connected FFN layer in the hybrid ViT-CNN architecture (Mamba Kabala *et al*., 2023). During the training process, batch normalization would adjust the activations of these layers by normalizing them according to the statistics calculated from each mini-batch. Batch normalization is an approach that standardizes the following algorithm.

```
def create_model(img_shape): # Define the model
inputs = layers.Input(shape=img_shape)
    x = layers.Conv2D(32, 3, activation="relu")(inputs) # Convolutional block with
    Batch Norm
```

```
    x = layers.BatchNormalization()(x)
    x = layers.MaxPooling2D(pool_size=(2, 2))(x)
    x = layers.Conv2D(64, 3, activation="relu")(x)
    x = layers.BatchNormalization()(x)
    x = layers.MaxPooling2D(pool_size=(2, 2))(x)
    x = layers.Flatten()(x) # Flatten layer
    x = layers.Dense(128, activation="relu")(x) # Fully connected layers
    x = layers.Dropout(0.5)(x)
outputs = layers.Dense(3, activation="softmax")(x) # Multi-class for healthy/diseased/
other
model = models.Model(inputs=inputs, outputs=outputs)
return model
model = create_model((224, 224, 3)) # Replace with your image size # Compile the
model
model.compile(optimizer="adam", loss="categorical_crossentropy",
metrics=["accuracy"])
# Train the model (Coffee Wilt disease and Coffee berry disease)
X_train, X_test, y_train, y_test = ... # Load your training and testing data
model.fit(X_train, y_train, epochs=10, validation_data=(X_test, y_test))
loss, accuracy = model.evaluate(X_test, y_test) # Evaluate the model
print("Test accuracy:", accuracy)
# Use the model for prediction
new_image = ... # Load a new image
prediction = model.predict(np.expand_dims(new_image, axis=0))
predicted_class = np.argmax(prediction) # Get the class with highest probability
```

In neural networks, the activations of each layer, excluding the input layer, are typically normalized across each mini-batch of training data to have a mean of zero and a standard deviation of one. This normalization enhances both the reliability and accuracy of the model during training. Batch normalization, as described by Chai Abel *et al*. (2025), is a technique that stabilizes the learning process, thereby accelerating training and improving convergence of the network. This approach mitigates the issue of internal covariate shift, a phenomenon that contributes to the vanishing or exploding gradient problem, which can hinder the effective training of deep neural networks.

Feature fusion and classification head

Feature fusion refers to the integration of data from multiple locations within the network. In the hybrid ViT-CNN architecture, the primary feature streams comprise convolutional feature maps generated by convolutional layers, which effectively capture local features such as edges and textures (Palanisamy *et al*., 2023). In parallel, the ViT module produces embeddings that represent global relationships across the entire leaf image. While convolutional features provide fine-grained, localized

information about specific regions of the leaf, the ViT embeddings capture broader disease patterns and their spatial context.

The final component of the disease prediction network is the classification head, which receives the fused features from preceding stages. This classification head typically consists of fully connected layers equipped with SoftMax or other non-linear activation functions. The number of neurons in the final layer corresponds to the number of target classes to be identified (i.e., healthy, rust disease, leaf spot disease). During training, the classification head learns to map the fused features to the corresponding disease probabilities (Balasundaram *et al*., 2025).

Counterfactual recommendation using hybrid ViT-CNN

The recommendations are hypothetical by comparing the actual generated data to historical data (Figure 2). There are three methods used to compare the suggestion. The dataset is divided into three types: historical data, current data, and a combination of current and historical data.

A coffee farmer utilizes a smartphone application integrated with a coffee plant health monitoring system to detect anomalies in plant components, including leaves, blossoms, and cherries. Upon observing irregularities, the farmer captures an image of the affected leaves and uploads it via the mobile application (Ratanoo *et al*., 2024). The image is transmitted to a server, where it is analyzed by the Affogato system, which utilizes a hybrid ViT-CNN model that leverages both global and local image features to identify the most probable disease.

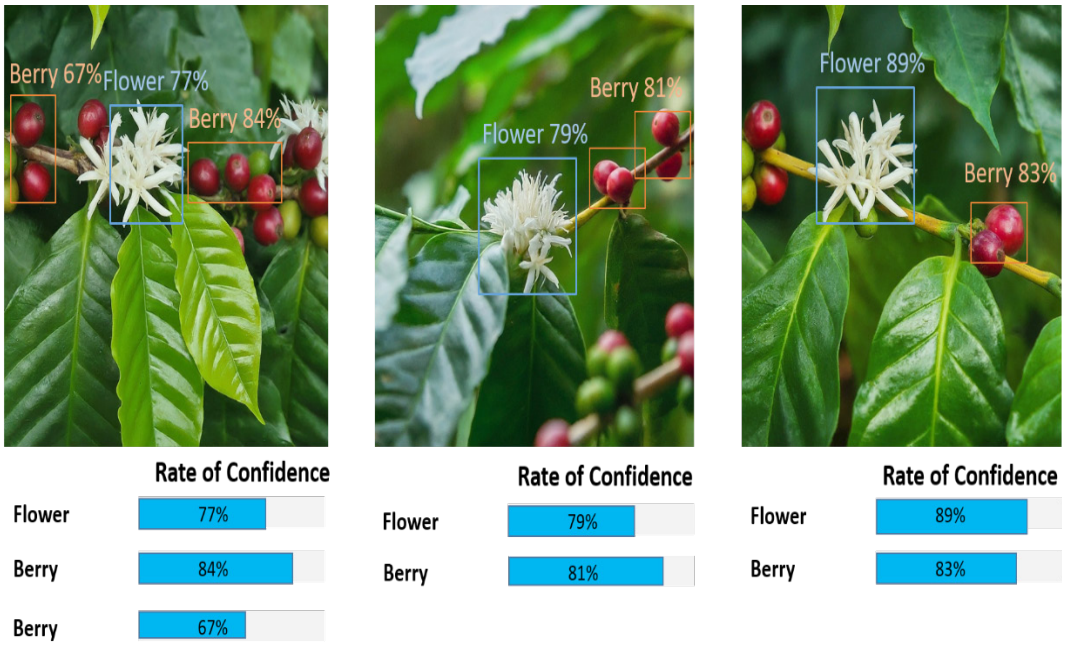

Figure 2. Virtual observation of the object detection results. Coffee flower and berry localization with confidence scores.

Subsequently, the system queries its database to retrieve detailed information regarding the diagnosed condition. The coffee producer is then presented with a comprehensive explanation of the disease and its associated symptoms, as well as treatment options. These treatment recommendations include assessments of therapeutic effectiveness and potential adverse effects. The application displays the identified disease and suggests the most appropriate counterfactual treatment, derived from the model's prediction and database insights (Selvanarayanan *et al*., 2024).

In the image analysis process, local binary patterns (LBPs) are used to examine localized texture features. LBPs function by detecting spatial variations in pixel intensity between a central pixel and its neighboring pixels. These intensity patterns are encoded into binary sequences, which represent the local texture characteristics (Equation 13).

$$LBP = \sum \left(2^n * S(p_i - p_c)\right) \tag{13}$$

where p_i is the intensity value of the i-th neighbor in the circular neighborhood, p_c is the intensity value of the central pixel, $S(x)$ is a Thresholding function, $S(x) = 1$ if $x \geq 0$, otherwise $S(x) = 0$, and n is the position of the bit in the binary string (0 for top-left neighbor, increasing clockwise). Round neighborhoods surround a central pixel in the image. This neighborhood contains 8 or 16 circular pixels. The intensity of the core pixel is compared to its neighbors in the defined neighborhood (Selvanarayanan *et al*., 2023). The binary code generation for each neighbor's intensity is designated as one if it exceeds the center pixel, and zero otherwise. Starting with the top-left neighbor, these binary values rotate clockwise. This binary string surrounds the central pixel and reflects the local texture pattern. During the conversion, each bit is assigned a weight of 2^n, where n represents the bit position in the binary string, starting from zero. The decimal value is calculated by summing the set bit weights.

RESULTS AND DISCUSSION

The proposed technique was tested using a dataset of coffee blossoming, coffee berry, and diseased leaf images. The hybrid ViT-CNN algorithm, combined with Vision, was used to detect coffee plant diseases at an early stage. Evaluation metrics such as accuracy, precision, and recall were used. The ViT-CNN model outperformed other current models, demonstrating higher levels of accuracy.

Evaluation setup and datasets

A total of 1056 images were collected using unmanned aerial vehicles (UAVs), fixed plantation cameras, and high-resolution agricultural imaging sensors. These images captured various stages of disease affecting the leaves, cherries, and roots of coffee plants. To ensure the generation of high-quality labeled data, expert agronomists

annotated the images during the selection process. Subsequently, data augmentation techniques were employed to enhance model generalization.

The proposed model was developed using the Keras Python framework (Version 2.7) and implemented in Python 3.6.5. It was executed on a workstation equipped with an Intel Core i5-8600k processor, a GeForce 1050Ti graphics card (4 GB memory), 16 GB of RAM, a 250 GB solid-state drive (SSD), and a 1 TB hard disk drive (HDD). The model was trained with the following hyperparameters: batch size of 5, learning rate of 0.01, dropout rate of 0.5, and 55 training epochs (Serrato-Diaz *et al*., 2024).

The input image sizes used in testing ranged from 32 × 32 × 3 to 256 × 256 × 3. Notably, the proposed method outperformed alternative approaches when applied to images with dimensions of 224 × 224 × 3. Model evaluation was conducted using standard classification metrics, including true positives, true negatives, false positives, and false negatives. A high-resolution digital camera was utilized to detect color variations during berry development stages, capture detailed images of coffee flowers at various blooming phases, identify sunken berries, and diagnose specific foliar diseases through changes in leaf pigmentation (Figure 3).

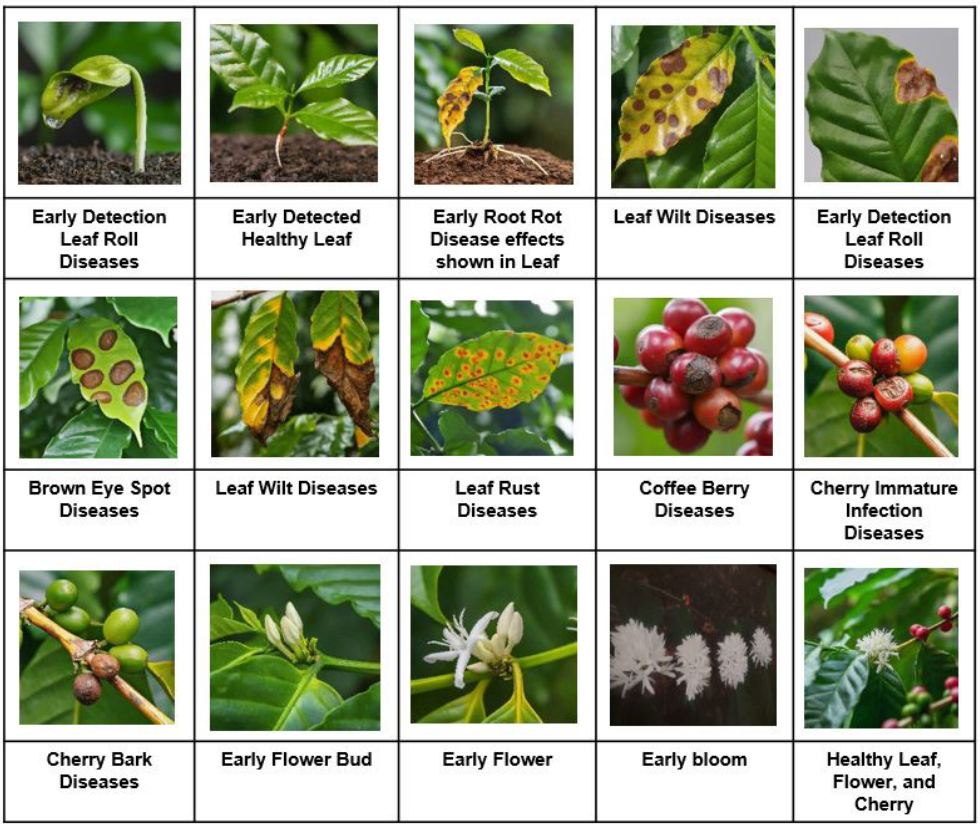

Figure 3. Visual representation of the 15-class coffee plant image dataset, showing representative samples from each class, encompassing various growth stages and disease manifestations used for training the convolutional neural network for automated classification.

The enlargement features for taking close-up pictures of a cherry, a berry, and a sickly leaf, data analysis, algorithm execution, and image processing were all handled by cloud computing, offering a large amount of storage space for pictures, annotations, and results (Sharma *et al*., 2024). The image processing software used Fiji and OpenCV. Custom algorithms for the early identification and surface segmentation of plant diseases were devised and executed using TensorFlow and PyTorch.

Field setup for observing coffee plant disease

A high-resolution digital camera with interchangeable lenses capable of capturing detailed photographs of coffee plant leaves, coffee berries, root rot, damping off disease, flowers at various stages of flowering, and close-up shots of flowers. Macro lenses help take close-up photos of individual cherries, damping disease, blossoms, and basic roots. The monitoring of coffee plant disease can be accomplished by using cloud storage systems to store collected photos (Figure 4). A virtual environment simulates real-world scenarios. The findings stemmed from the implemented system following the training and execution of a smartphone application designed to detect coffee plant diseases. A 2.4 GHz TM i5-9300H CPU, Google Collab, and Python 3.0.6 were used to train and execute a smartphone application for identifying coffee plant diseases (Signo *et al*., 2024).

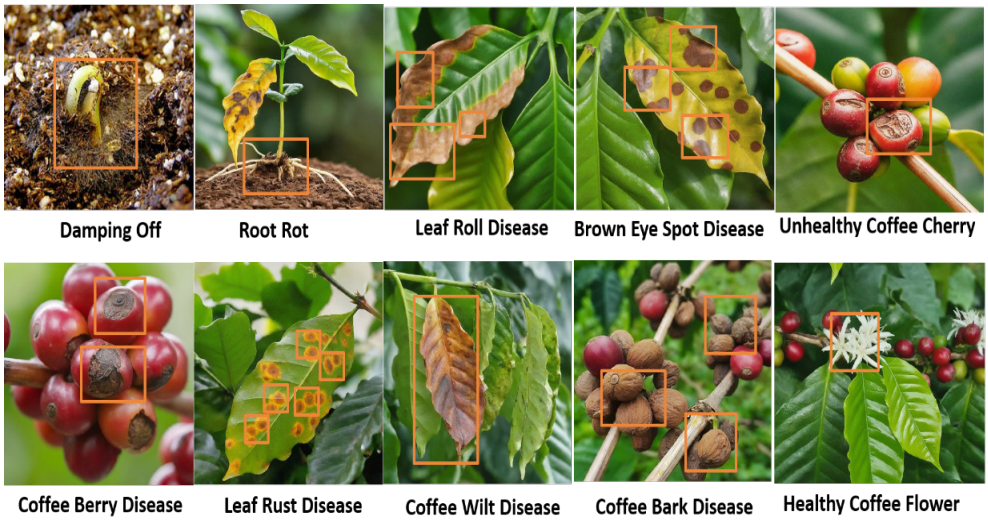

Figure 4. Real-time coffee plant disease monitoring via object detection.

Performance evaluation using hybrid ViT-CNN

Classification accuracy (Equation 14) is a metric that represents the percentage of samples correctly classified as healthy or sick. True Positives (TP) indicate several accurately diagnosed diseased samples. True Negatives (TN) show several correctly categorized healthy samples. Total Samples is a list of all the samples in the dataset.

The precision (Equation 15) statistic calculates the proportion of correctly identified unhealthy cases while minimizing false positives, indicating the accuracy of positive predictions (Taha and Hussein, 2022).

$$Accuracy = \frac{True\ Positives\ +\ True\ Negatives}{Total\ Samples} \tag{14}$$

$$Precision = \frac{TP}{TP\ +\ False\ Positives\ (FP)} \tag{15}$$

False positives (FP) are the number of healthy samples that are incorrectly classified as sick. The percentage of actual disease instances that are successfully recognized with no false negatives is known as recall (sensitivity) (Equation 16). False Negatives (FN) are samples of sick material that are mistaken for healthy. The F1-Score (Equation 17) effectively balances precision and recall, making it especially useful when both metrics are required. The intersection over union (IoU) segmentation metric (Equation 18) assesses the degree of overlap between actual and projected disease zones on coffee plant leaves, cherries, flowers, or fruits.

$$Recall = \frac{TP}{TP\ +\ False\ Negatives\ (FN)} \tag{16}$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{17}$$

$$IoU = \frac{Intersection\ Area}{Union\ Area} \tag{18}$$

The area under the curve (AUC) measures the model's ability to differentiate between healthy and diseased groups using two metrics: true positive rate (TPR) (Equation 19) and false positive rate (FPR) (Equation 20). These metrics can be used for multi-class tasks, such as understanding various types of sickness, by assessing each class separately. Crossing the line where the masks of the expected and ground reality overlap. The entire union area is covered by either the expected or actual mask, whichever is larger.

$$TPR = \frac{TP}{TP\ +\ FN} \tag{19}$$

$$FPR = \frac{FP}{FP\ +\ TN} \tag{20}$$

True positive (TP) represents the number of correctly classified positive cases where indicated. False negative (FN) refers to several cases that were misclassified as negative (missed positive). False positive (FP) count of cases misclassified as negative but actually positive. True negative (TN) refers to several accurately identified undesirable events (Tamilvizhi *et al*., 2022). The experimental parameters were set to 16 batches with a learning rate of 0.01, and the performance was compared to other deep-learning models for detecting coffee plant diseases (Table 1).

Table 1. Comparative performance analysis of hybrid vision transformer-convolutional neural network (ViT-CNN) model and existing models, showing accuracy and loss metrics for the proposed model across training, validation, and testing phases, benchmarked against state-of-the-art deep learning models.

Model	Training accuracy	Validation accuracy	Testing accuracy	Training loss	Validation loss	Testing loss
Hybrid ViT-CNN	1.000	0.9867	0.9879	0.0047	0.0124	0.0578
MVGG16	0.9976	0.9802	0.9711	0.0720	0.0777	0.0701
Inception V3	0.9941	0.9791	0.9766	0.0752	0.0711	0.0714
Xception	0.9921	0.9716	0.9657	0.0272	0.0930	0.0817
DenseNet-121	0.9711	0.9651	0.9611	0.0547	0.0577	0.0991
MobileNet-V2	0.9418	0.9374	0.9457	0.0478	0.0321	0.0840
Visual transformer (ViT)	0.9089	0.8947	0.9055	0.0145	0.0477	0.0944
ResNet50	0.8776	0.8624	0.8518	0.0594	0.0749	0.0741

Pre-trained models utilized in this study include MVGG16, Inception V3, Xception, DenseNet-121, MobileNet-V2, ViT, and ResNet50. Among these, the hybrid ViT-CNN model demonstrated the most favorable overall performance across key evaluation metrics. Specifically, it achieved a high training-validation accuracy balance (0.9867), along with superior classification outcomes, including accuracy (0.9881), precision (0.9893), recall (0.9895), and area under the curve (AUC) (0.9896). These results indicate the model's strong capability in both the accurate classification of coffee plant diseases and the effective differentiation between healthy and diseased specimens.

The intersection over union (IoU) value of 0.9833 indicates a high degree of agreement between the predicted and actual disease regions (Table 2). The proposed model (blue line) achieves a higher training accuracy, approaching 1.0, and does so more rapidly than the previous model (green line), demonstrating superior learning efficiency (Figure 5). Furthermore, the validation accuracy (dashed blue line) closely follows the training accuracy, exhibiting a narrower gap in comparison to the existing model (dashed green line) (Figure 6). This suggests improved generalization performance of the new model. Finally, the classification-based confidence score for specific disease detection (Figure 7) is defined as the predicted probability of belonging to the corresponding disease class (Equation 21).

Table 2. Comprehensive performance metrics of the hybrid vision transformer-convolutional neural network (ViT-CNN) comparing accuracy, precision, F1-score, recall, area under the curve (AUC), and intersection over union (IoU) against other deep learning models, demonstrating superior performance across multiple evaluation metrics.

Model	Accuracy	Precision	F1-score	Recall	AUC	IoU
Hybrid ViT-CNN	**0.9881**	0.9893	0.9849	0.9895	0.9896	0.9833
MVGG16	0.9679	0.9643	0.9604	0.9691	0.9616	0.9619
Inception V3	0.9517	0.9515	0.9521	0.9536	0.9604	0.9607
Xception	0.9257	0.9280	0.9251	0.9232	0.9225	0.9257
DenseNet-121	0.8871	0.8793	0.8814	0.8890	0.8891	0.8879
MobileNet-V2	0.8557	0.8505	0.8561	0.8536	0.8644	0.8657
Visual Transformer (ViT)	0.8152	0.8182	0.8106	0.8132	0.8125	0.8157
ResNet50	0.7857	0.7882	0.7856	0.7832	0.7820	0.7807

$$Confidence_{score}(Disease_i) = Predicted_{probability}(Class_i) \quad (21)$$

where $Disease_i$ represents a specific coffee plant disease class, and $Predicted_{probability}(Class_i)$ is the probability value predicted by the model for the image region belonging to class.

The model demonstrated exceptional accuracy for detecting critical objects (Figure 7), such as "Bud" (99.99 %), "Plant" (98.67 %), and "Tree" (97.23 %), indicating strong reliability in identifying plant structures. The horizontal bar chart efficiently displays the confidence scores, making it easy to compare detection accuracy across different objects.

The bubble-enhanced line graph adds an interesting visual layer by displaying confidence variations in a structured manner (Figure 8). High confidence indicates that the model is well-trained and capable of distinguishing between various plant components with minimal error.

Mobile application

Farmers can utilize the user-friendly Affogato mobile application to upload images of their coffee plants, including diseased flowers, cherries, and leaves (Figure 9). Upon upload, each image undergoes a series of preprocessing steps, such as resizing, normalization, and, if necessary, color space adjustments, to ensure compatibility with the subsequent analytical model. The preprocessed image is then input into the hybrid ViT-CNN model, which has been trained on an extensive dataset comprising coffee plant images under various conditions, including healthy and diseased specimens of flowers, cherries, and leaves.

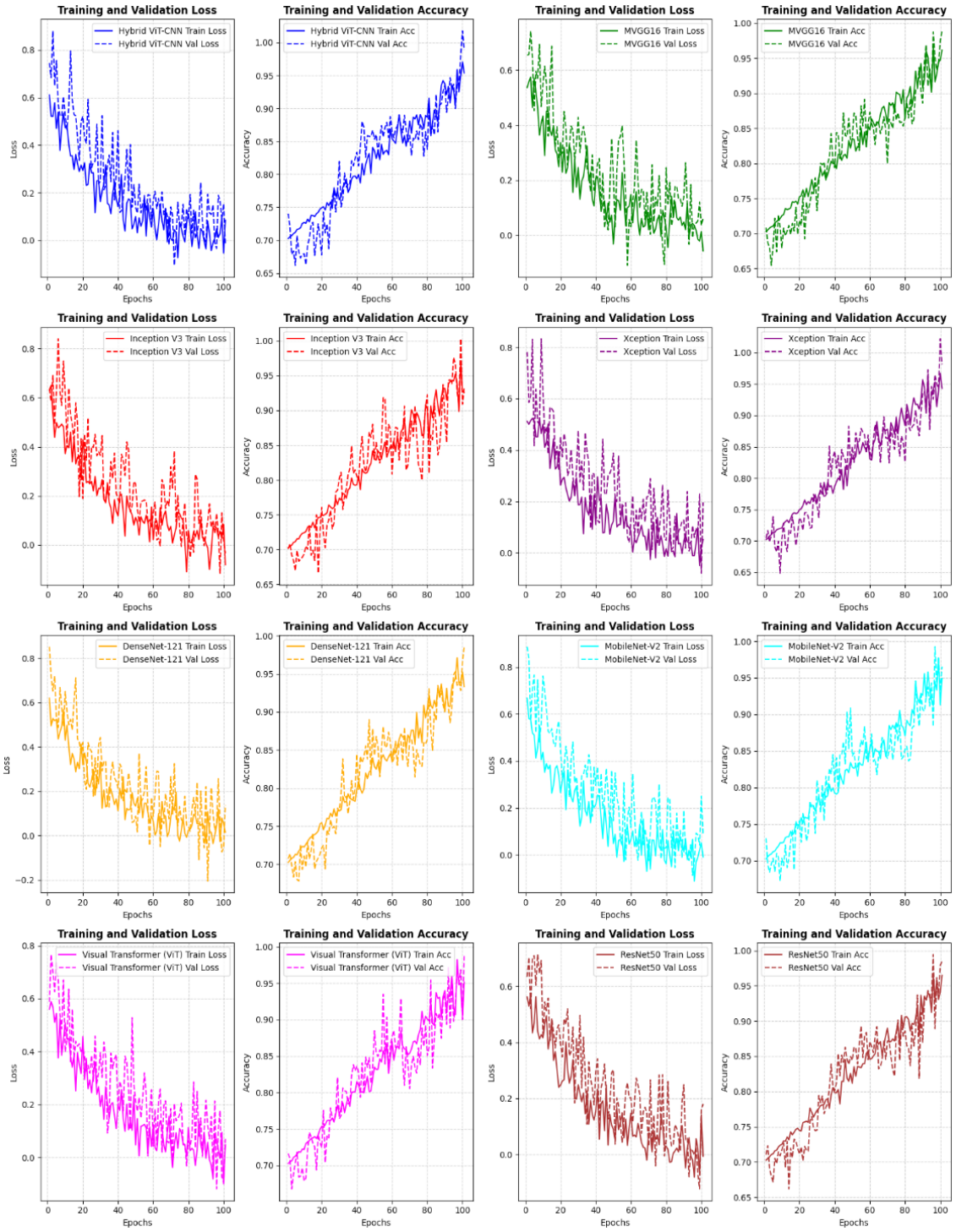

Figure 5. Comparison of training and validation loss and accuracy for different deep learning models over 100 epochs.

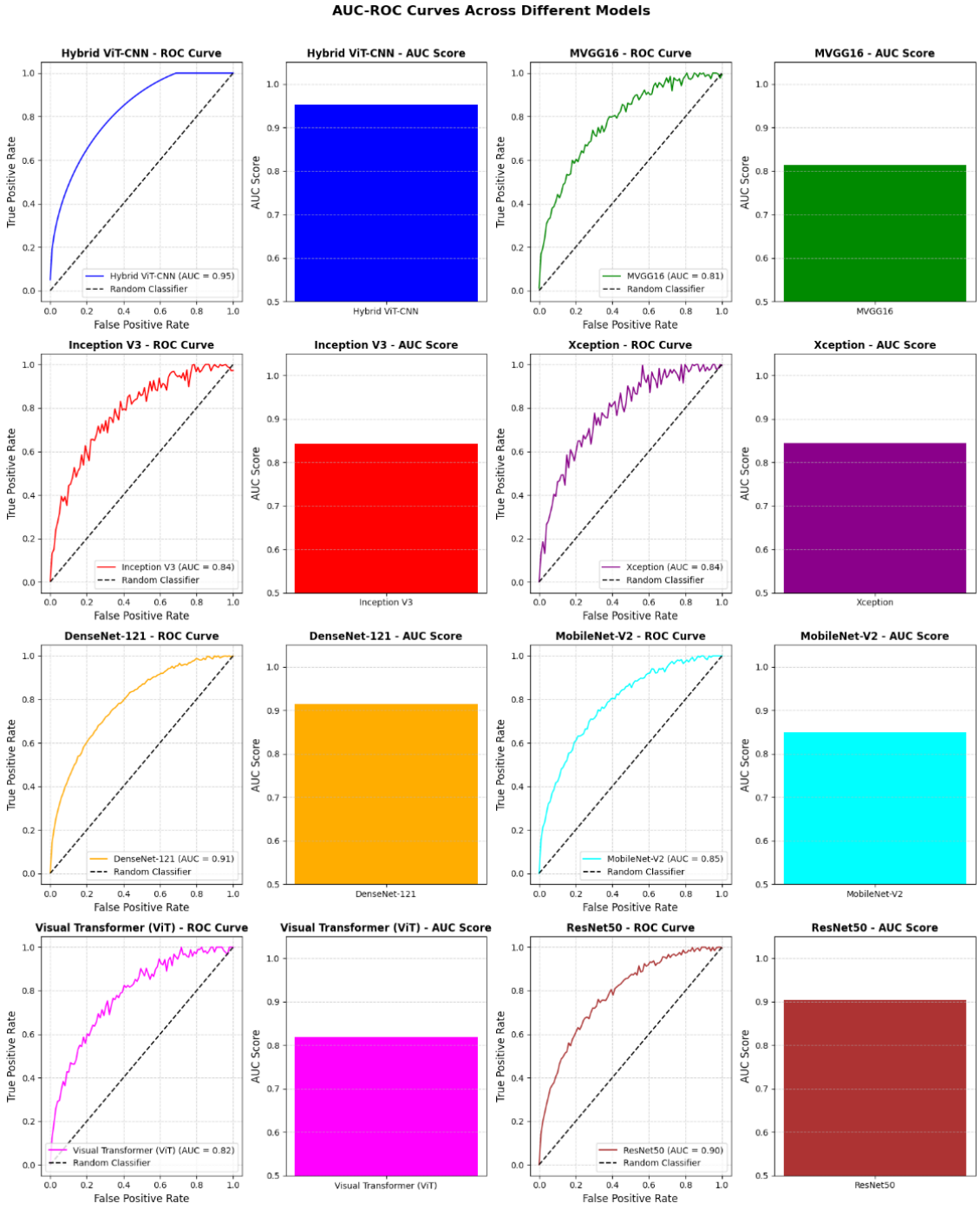

Figure 6. Performance comparison of the area under the curve of the receiver operating characteristic (AUC-ROC) scores for each model.

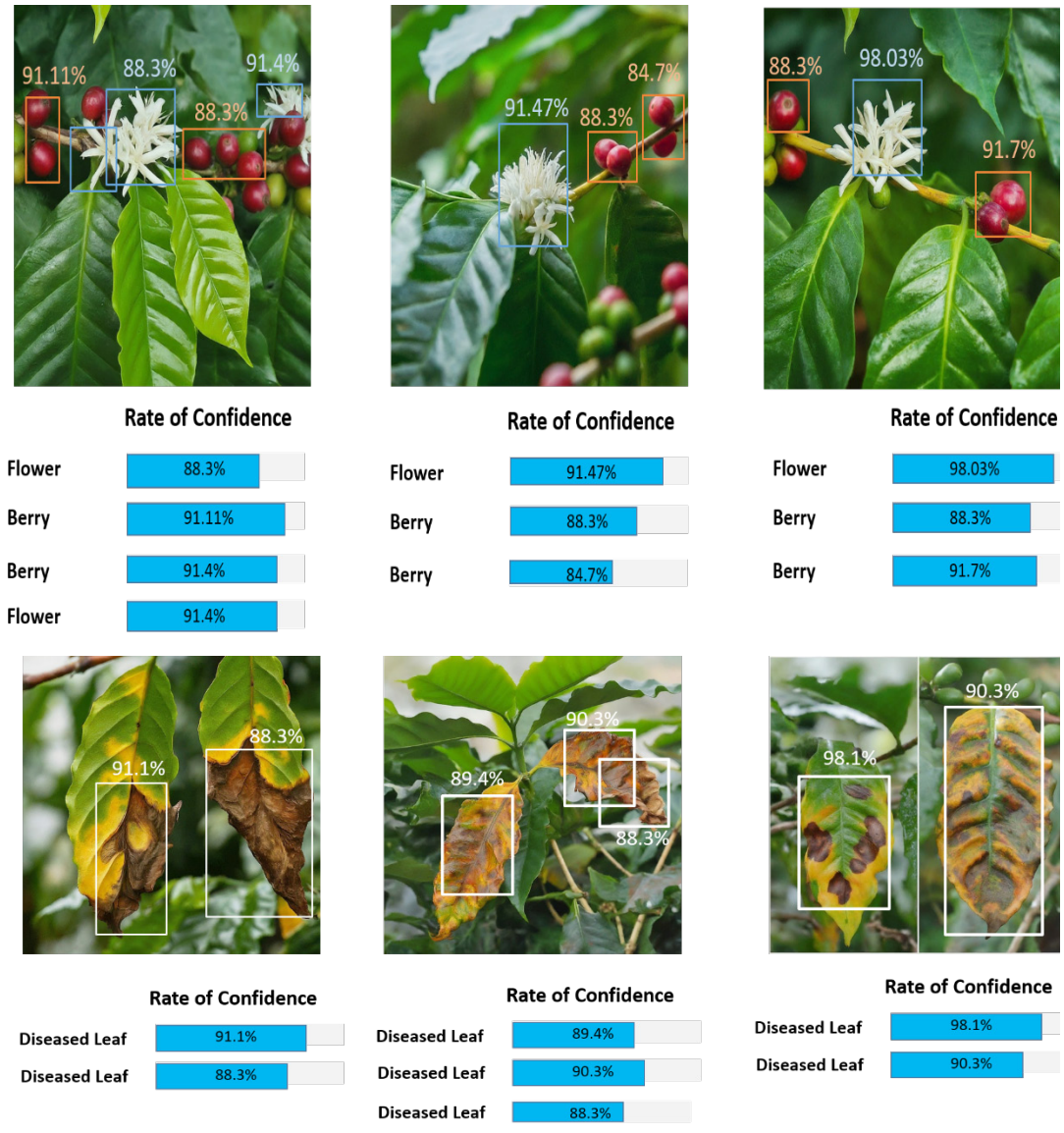

Figure 7. Coffee plant disease detection using confidence rates.

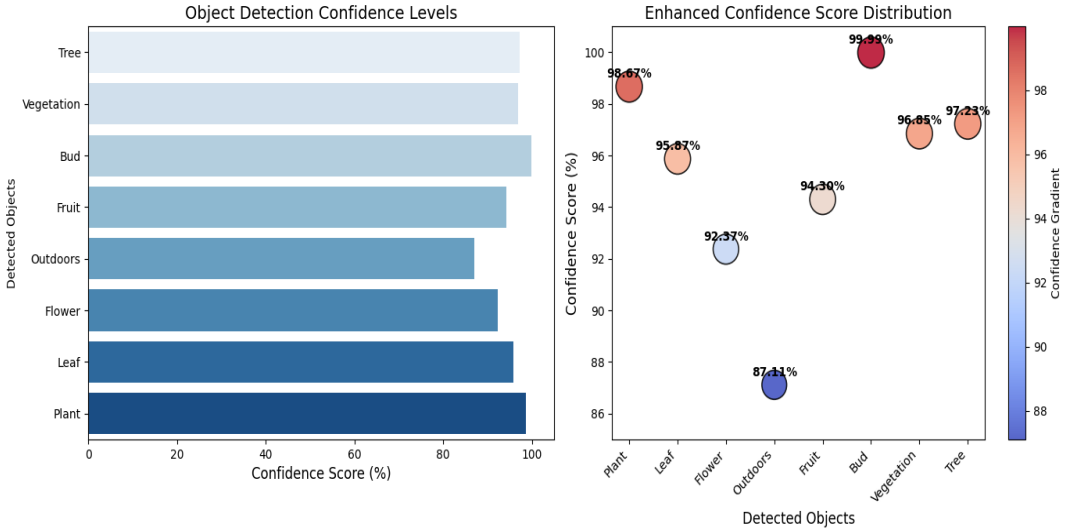

Figure 8. Visualization of object detection confidence levels. A comparative analysis is presented using two graphical formats: a bar chart (left) and a bubble chart (right). The bar chart illustrates the confidence scores associated with detected objects, while the enhanced bubble chart displays the distribution of these scores using variations in both size and color gradients to convey magnitude and intensity.

Figure 9. Affogato: an AI-powered system for the diagnosis and prevention of coffee plant diseases. This platform provides a step-by-step framework that guides farmers from the initial detection of plant diseases to the implementation of preventive strategies, utilizing counterfactual analysis and farm-centric advisory approaches.

The model performs real-time analysis to identify the specific plant part depicted (i.e., flower, cherry, or leaf) and to predict the most probable condition based on visual characteristics. Subsequently, the system generates personalized recommendations informed by the predicted condition and supplemented by additional contextual data provided by the farmer. This contextual data may include historical and current observations, such as geographical location, soil type, and climatic conditions.

CONCLUSIONS

A smartphone-based system for the detection of plant diseases and the recommendation of treatments was developed using machine learning techniques. The proposed hybrid model, integrating Vision Transformers and Convolutional Neural Networks (ViT-CNN), combines the global contextual awareness of Vision Transformers with the localized feature extraction capabilities of Convolutional Neural Networks. This synergistic architecture enables accurate diagnosis of various coffee plant diseases. A key advancement in the system is the implementation of counterfactual suggestion processes, which extend beyond mere disease identification. These processes simulate the health status of the plant under multiple hypothetical treatment scenarios, thereby allowing farmers to explore "what-if" analyses and make more informed decisions regarding disease management. Future research should focus on enhancing the model's generalizability across diverse coffee plant varieties, environmental conditions, and disease phenotypes.

ACKNOWLEDGEMENTS

Data related to the study are publicly available and can be accessed at Selvanarayanan R. 2024. Coffee plant leaf disease. Zenodo. https://zenodo.org/records/13674477

The source code used for leaf disease classification is available at Selvanarayanan R. 2024. Leaf disease classification code. Zenodo. https://zenodo.org/records/11107971

REFERENCES

Abdulsahib GM, Hassan HJ, Khalaf OI. 2024. A modified bandwidth prediction algorithm for wireless sensor networks. Journal of Information Science and Engineering 40 (1): 177–188. https://doi.org/10.6688/JISE.202401_40(1).0011

Alharbi M, Rajagopal SK, Rajendran S, Alshahrani M. 2023. Plant disease classification based on ConvLSTM U-net with fully connected convolutional layers. Traitement du Signal 40 (1): 157. https://doi.org/10.18280/ts.400114

Anand D, Khalaf OI, Abdulsahib GM, Chandra GR. 2024. Original research article identification of meningioma tumor using recurrent neural networks. Journal of Autonomous Intelligence 7 (2): 1–27. https://doi.org/10.32629/jai.v7i2.653

Balasundaram A, Sundaresan P, Bhavsar A, Mattu M, Kavitha MS, Shaik A. 2025. Tea leaf disease detection using segment anything model and deep convolutional neural networks. Results in Engineering 25: 103784. https://doi.org/10.1016/j.rineng.2024.103784

Chai AYH, Lee SH, Tay FS, Goëau H, Bonnet P, Joly A. 2025. PlantAIM: A new baseline model integrating global attention and local features for enhanced plant disease identification. Smart Agricultural Technology 10: 100813. https://doi.org/10.1016/j.atech.2025.100813

Chaudhari RR, Jain S, Gupta S. 2025. Agricultural machine learning platform: Enhancing crop suggestion and crop yield estimates. Journal of Integrated Science and Technology 13 (1): 1017. https://doi.org/10.62110/sciencein.jist.2025.v13.1017

Chinnasamy P, Wong WK, Raja AA, Khalaf OI, Kiran A, Babu JC. 2023. Health recommendation system using deep learning-based collaborative filtering. Heliyon 9 (12): 1–27. https://doi.org/10.1016/j.heliyon.2023.e22844

Dias JS, Boa Sorte LX, Fambrini F, Saito JH. 2025. Coffee plant disease detection using JSEG segmentation and near sets clustering. Fifth Symposium on Pattern Recognition and Applications (SPRA 2024) 13540: 40–52. https://doi.org/10.1117/12.3056434

Duhan S, Gulia P, Gill NS, Narwal E. 2025. RTR_Lite_MobileNetV2: A lightweight and efficient model for plant disease detection and classification. Current Plant Biology 42: 100459. https://doi.org/10.1016/j.cpb.2025.100459

Essa HM, Murshid AM. 2023. Optimizing image processing with CNNs through transfer learning: Survey. Al-Kitab Journal for Pure Sciences 7 (1): 57–68. https://doi.org/10.32441/kjps.07.01.p6

Isinkaye FO, Erute ED. 2022. A smartphone-based plant disease detection and treatment recommendation system using machine learning techniques. Trees 10 (1): 1–18. https://doi.org/10.14738/tmlai.101.11313

Jiang J, Ji H, Zhou G, Pan R, Zhao L, Duan Z, Liu X, Yin J, Duan Y, Ma Y, *et al*. 2025. Non-destructive monitoring of tea plant growth through UAV spectral imagery and meteorological data using machine learning and parameter optimization algorithms. Computers and Electronics in Agriculture 229: 109795. https://doi.org/10.1016/j.compag.2024.109795

Jung M, Song JS, Shin AY, Choi B, Go S, Kwon SY, Park J, Park SG, Kim YM. 2023. Construction of deep learning-based disease detection model in plants. Scientific Reports 13 (1): 7331. https://doi.org/10.1038/s41598-023-34549-2

Lee CH, Rianto B. 2024. An AI-powered e-nose system using a density-based clustering method for identifying adulteration in specialty coffees. Microchemical Journal 197: 109844. https://doi.org/10.1016/j.microc.2023.109844

Mamba Kabala D, Hafiane A, Bobelin L, Canals R. 2023. Image-based crop disease detection with federated learning. Scientific Reports 13 (1): 19220. https://doi.org/10.1038/s41598-023-46218-5

Palanisamy S, Abdulsahib GM, Khalaf OI, SS A Wong WK, Pan SH. 2023. Design of artificial magnetic conductor based stepped V-shaped printed multiband antenna for wireless applications. International Journal of Advances in Soft Computing and its Applications 15 (3): 1–25.

Ratanoo R, Walia SS, Saini KS, Dheri GS. 2024. Residual effects of chemical fertilizers, organic manure, and biofertilizers applied to preceding Gobhi Sarson crop on summer mung bean (*Vigna radiata* L.). Legume Research 47 (1): 64–68. https://doi.org/10.18805/lr-4767

Rehman M, Petrillo A, Baffo I, Iovine G, de Felice F. 2025. Optimizing coffee supply chain transparency and traceability through mobile application. Procedia Computer Science 253: 2116–2126. https://doi.org/10.1016/j.procs.2025.01.272

Selvanarayanan R, Rajandran S, Alotaibi Y. 2023. Using hierarchical agglomerative clustering in e-nose for coffee aroma profiling: Identification, quantification, and disease detection. Instrumentation Mesure Métrologie 22 (4): 127–140. https://doi.org/10.18280/i2m.220401

Selvanarayanan R, Rajendran S, Alotaibi Y. 2024. Early detection of disease in coffee cherry based on computer vision techniques. Computer Modeling in Engineering and Sciences 139 (1): 759–782. https://doi.org/10.32604/cmes.2023.044084

Serrato-Diaz LM, Mariño YA, de Jesús-González J, Goenaga R, Bayman P. 2024. Coffee fruit rot: The previously unrecognized role of *Fusarium* and its interactions with the coffee berry borer (*Hypothenemus hampei*). Phytopathology 22 (1): 1–18. https://doi.org/10.1094/phyto-02-24-0046-r

Shafik W, Tufail A, de Silva LC, Apong RAAHM. 2025. A novel hybrid inception-xception convolutional neural network for efficient plant disease classification and detection. Scientific Reports 15 (1): 3936. https://doi.org/10.1038/s41598-024-82857-y

Shafik W, Tufail A, de Silva LC, Apong RAAHM. 2025. An enhanced deep convolutional neural network for plant disease detection and classification: Elevating sustainable agriculture. Artificial Intelligence and Data Science for Sustainability 12: 297–322. https://doi.org/10.4018/979-8-3693-6829-9.ch010

Sharma V, Tripathi AK, Mittal H, Nkenyereye L. 2025. SoyaTrans: A novel transformer model for fine-grained visual classification of soybean leaf disease diagnosis. Expert Systems with Applications 260: 125385. https://doi.org/10.1016/j.eswa.2024.125385

Signo SDR, Tuquero CLG, Arboleda ER. 2024. Coffee disease detection and classification using image processing: A literature review. International Journal of Science and Research Archive 11 (1): 1614–1621. https://doi.org/10.30574/ijsra.2024.11.1.0212

Taha MD, Hussein KA. 2022. Generation S-box, and P-layer for PRESENT algorithm based in 6D hyper chaotic system. Al-Kitab Journal for Pure Sciences 7 (1): 48–56. https://doi.org/10.32441/kjps.07.01.p5

Tamilvizhi T, Surendran R, Anbazhagan K, Rajkumar, K. 2022. Quantum behaved particle swarm optimization-based deep transfer learning model for sugarcane leaf disease detection and classification. Mathematical Problems in Engineering 2022 (1): 3452413. https://doi.org/10.1155/2022/3452413

Wang Y, Wang Y, Mu J, Raza Mustafa G, Wu Q, Wang Y, Zhao B, Zhao S. 2025. Enhanced multiscale plant disease detection with the PYOLO model innovations. Scientific Reports 15 (1): 5179. https://doi.org/10.1038/s41598-025-89034-9