Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

Dual-Path Attention Fusion Network with Adaptive Quantum Monarch Butterfly Optimization for Banana Plant Disease Detection

Kanimalar C.[1], Karthikeyan M.[2]

[1, 2]Department of Computer and Information Science, Annamalai University, Chidambaram, Tamilnadu, India.

E-mail: [1]ckani.chelliah@gmail.com, [2]karthiaucse@gmail.com

Abstract

Diagnosis of banana plant disease is a crucial aspect of sustaining the harvest of crops and their quality. Visual inspection of certain diseases like Black Sigatoka, Panama disease, and aphids is not easy and can lead to misjudgments. Generally, traditional deep learning approaches have been previously used but they have not performed well in addressing issues of class imbalance, sensitive disease differentiation and noisy images obtained in the field. Furthermore, most models are based on a collection of predetermined preprocessing methods and single-path networks that limit their ability to generalize to a wide variety of environments. Current methods of deep learning tend to achieve reasonable overall performance but fail to perform well on key performance indicators such as recall and F1-score when considering underrepresented and overlapping classes, such as Yellow and Black Sigatoka. Such constraints impede efficient field implementation, as diseases of minority classes are often falsely classified. To overcome these deficiencies, we develop a novel Duel-Path Attention Fusion Network (DPAFNet) that is trained utilizing adaptive quantum monarch butterfly optimization (AQMBO). The concept behind the proposed model is to feed MaxViT and HorNet-S two feature extractors to deliver global contextual details and minute-scale textural features. The traditional filters which do a reasonable job in handling dynamic noise and contrast are replaced by a learnable preprocessing unit. The cross-layer fusion attention encourages interclass discriminative learning of diseased plants. The suggested model has been trained and tested on an open-source dataset of Mendeley banana disease, which includes 5,170 images in 7 disease categories and 1 control condition. The accuracy, F1-score and MCC of 98.6% and 0.93 and 0.87 respectively (achieved experimentally) demonstrate the superiority of DPAFNet over baseline models such as EfficientNetB0 (accuracy 95.0%), DenseNet121 and ResNet50 (accuracy 93.50% and 92.0% respectively). As can be seen, the model had a 0.26-0.48 increase in F1-score in the challenging Panama disease category. These results prove that the proposed architecture can be successfully used to achieve high-accuracy disease classification in smart agriculture that is robust and prepared for field implementation.

Keywords: Dual-Path Attention Fusion Network (DPAFNet), Adaptive Quantum Monarch Butterfly Optimization (AQMBO), MaxViT; HorNet-S, Cross-Layer Attention Fusion (CLAF), Banana Leaf Disease Classification.

DOI: https://doi.org/10.36548/jiip.2025.3.011
Received: 10.07.2025, received in revised form: 14.08.2025, accepted: 30.08.2025, published: 12.09.2025

1. Introduction

Banana (Musa spp.) is one of the most widely consumed fruits across the globe, and it is a highly significant food security and economic livelihood crop in the tropics. It has application not only in the food industry but also in the textile, pharmaceutical and feed industries. However, banana production is plagued by many pests and diseases. Black Sigatoka and Panama Disease candrastically reduce yield and quality [1]. Hence, it is important that these conditions are identified at an early stage and with preciseness so that the transmission of these diseases can be prevented, and plans for controlling them without harming the environment need to be put into effect. However, traditional manual diagnostic methods are error-prone, controversial regarding competence, and inadequate for large-scale monitoring [2].

The issues faced by field-based diagnosis such as heterogeneous imaging conditions, overlapping symptoms among diseases, and class imbalance in the data are challenging. Diseases like Yellow and Black Sigatoka exhibit nearly identical lesion patterns that make visual distinction more complex. Moreover, a number of disease groups do not have sufficient labeled data resulting in biased model training and poor generalization. Histogram equalization is an older preprocessing approach that is not suitable for a wide variety of field conditions [3]. Furthermore, the application of deep learning models on edge devices like smartphones or drones is not viable in real-time due to heavy computational requirements [4-5].

CNNs have been implemented in various automated disease detection procedures and have outperformed other approaches on structured image-based tasks [6]. Transfer and multiscale feature methods have enhanced models like VGG16 and ResNet [7-8]. However, they are limited by being based on local receptive fields to identify distributed disease features. Mobile Lightweight CNNs can be applied in mobile systems at the expense of classification accuracy. These limitations have resulted in a move toward stronger structures such as attention-based Transformers that can handle both local and global patterns.

Transformers such as Swin Transformer [9] and Vision Transformer (ViT) [10] function effectively to capture the long-range correlation between the appearance features of an observed disease. Swin's self-attention and ConvNeXt's convolutional precision provide better performance in disease detection through window and convolutional strategies respectively. Other attention-based models are not adaptively processed, lack multi-level feature integration, and do not tune their hyperparameters, which limits their real-world applicability in unstructured agriculture. All these limitations highlight the need for a self-adapting multi-path model that can react to varying image properties and class distribution.

The research work presented in this paper suggests DPAFNet, which is a Dual-Path Attention Fusion Network designed specifically to be optimal for banana disease classification. It begins with a CNN based trainable preprocessing module that adjusts dynamically for contrast and noise. It consists of two deep branches incorporated in parallel: MaxViT to learn global-local features and HorNet-S to understand fine texture processes. MaxViT will be introduced as a hybrid CNN-Transformer backbone that is aimed to learn global semantic patterns and long-range relations, and HorNet-S as a lightweight convolutional net dedicated to extracting fine-grained local textures. These explanations will increase readers' understanding of the reasons why both are combined in the dual-path architecture. In addition to augmenting classification accuracy, the new architecture improves interpretability with attention-guided learning. The preprocessing process highlights lesion contours and attenuates background interference, and the Cross-Layer Attention Fusion module emphasizes spatial patterns and channel patterns most associated with disease. These processes enable the predictions of the model to be followed back to observable leaf symptoms, rendering the decision-making process both transparent and actionable for farmers.

Their features are fused via a Cross-Layer Attention Fusion (CLAF) based on attention-based residual fusion. An Adaptive Quantum Monarch Butterfly Optimization (AQMBO) algorithm is also added to the proposed model for adaptive fine-tuning of learning rates and dropout rates through hybrid global-local search. Experimentation on a banana disease dataset supported the proposed DPAFNet in improving accuracy and generalizability compared to other methods.

The major contributions of this research work are as follows

1. Proposed a new multi-path classification model (DPAFNet) by parallelly combining MaxViT and HorNet-S to integrate global contextual patterns and local textural details for better disease classification.

2. Introduced a learnable preprocessing module that substitutes conventional filters and allows dynamic improvement of leaf images under different lighting and noise conditions.

3. Integration of a Cross-Layer Attention Fusion (CLAF) approach to efficiently combine dual-path features through channel and spatial attention, enhancing discrimination among similar disease symptoms.

4. Development of the Adaptive Quantum Monarch Butterfly Optimization (AQMBO) algorithm for self-adaptive hyperparameter adjustment, facilitating rapid convergence and better performance.

5. Large-scale validation on a publicly released Mendeley banana leaf disease dataset, with higher metrics like an accuracy of 98.6%, an F1-score of 0.93, and an MCC of 0.87. The performance of the proposed model surpasses that of EfficientNetB0, ResNet50, DenseNet121, VGG16 and ConvNextTiny models.

The rest of the discussions in the paper are discussed following the sequence. Section 2 describes the literature review of the latest research articles. Section 3 gives the mathematical model of the proposed disease detection model. Section 4 gives the results and discussion and section 5 concludes the research study.

2. Related Works

Recent advancements in plant disease detection have utilized deep learning models for improved accuracy and efficiency. Numerous studies have explored convolutional architectures, optimization techniques, and hybrid feature extraction strategies across various crops. This section reviews key contributions, methodologies, and limitations that motivate the development of the proposed DPAFNet model. The comparative analysis of deep learning techniques for the detection of plant diseases detection in [11] incorporates eight advanced models like Xception, ResNet15v2, DenseNet201, and variants of EfficientNet (B0 to B4). Experimentations on benchmark datasets demonstrate that EfficientNetB3 achieves the best performance with higher accuracy and minimal loss compared to other models. Models like EfficientNetB0 and B2 also performed well but lagged slightly in terms of loss values. While the experimental results validate the potential of deep learning in plant pathology, the study is limited by dataset class imbalance and a narrow crop focus.

A comprehensive banana disease identification model is presented in [12] by integrating a dual-method approach involving image denoising and advanced feature extraction. The methodology starts with the K-scale VisuShrink Algorithm (KVA), which enhances banana leaf images by applying adaptive wavelet-based thresholding to reduce noise while preserving edge details. Following preprocessing, the paper introduces GR-ARNet, a deep learning model that utilizes Ghost Modules, ResNeSt Modules, and a hybrid RReLU-Swish activation strategy atop a ResNet-50 backbone. This architecture is developed to handle subtle inter-class variations and extract fine-grained features. The model achieved better classification accuracy on a diverse dataset comprising diseases like Sigatoka, Cordana, and Pestalotiopsis. The presented model limitations include a focus on only three disease classes and potential computational overhead during real-time deployment.

Convolutional neural network (CNN) based early detection of banana diseases is reported in [13] specifically Fusarium Wilt and Black Sigatoka. The methodology centers around a four-layer CNN implemented using TensorFlow, designed to work on mobile platforms through TensorFlow Lite. The experimental results showed the model achieving best-case accuracy with corresponding improvements in precision and recall across disease categories. However, one of the significant limitations is its reduced inference performance when deploying the model using TensorFlow Lite, which affected detection precision in low-resource environments. A similar CNN based model presented in [14] is designed to detect three banana leaf diseases Pestalotiopsis, Sigatoka, and Cordana. The model incorporates a lightweight architecture built on the Fire module concept, optimized using Bayesian Optimization to fine-tune hyperparameters such as learning rate and regularization. Data augmentation techniques and transfer learning were applied to enhance model generalization on the BananaLSD dataset. Experimental evaluations demonstrate that BananaSqueezeNet achieves superior classification accuracy outperforming models like ResNet-101 and Inception-V3 while requiring significantly fewer computational resources. However, the model is limited by a relatively small dataset size and narrow disease diversity.

Another CNN based solution is reported in [15] for early detection of plant diseases using leaf images. The methodology involves a sequential CNN architecture composed of convolutional, pooling, and fully connected layers, preceded by robust data preprocessing steps such as normalization, standardization, and image augmentation. The model was trained on the Plant Village dataset and achieved an average classification accuracy. Experimentally, the model outperformed the traditional K-Nearest Neighbors (KNN) classifier, confirming the advantages of deep learning in handling complex visual patterns. However, its performance indicates its relatively low validation accuracy and introduces limitations in generalization.

The plant disease diagnostic model presented in [16] introduces a large-scale plant-specific pretraining methodology to enhance deep learning performance in plant disease recognition. Unlike conventional approaches that rely on general-purpose models trained on ImageNet, this proposed model constructs a domain-specific pretraining image set from different agricultural environments. The presented pre-trained CNN and Transformer-based models are trained on this dataset to improve classification, detection, and segmentation of plant diseases. Experimental evaluations demonstrate significant improvements in model accuracy and convergence speed, especially when combined with ImageNet weights. However, the main limitation lies in the computational cost of training such large models and the lack of mobile-optimized variants.

The Deep Spectral Generative Adversarial Neural Network (DSGAN2) presented in [17] detects diseases in plant leaves through spectral data analysis. The presented model incorporates an Improved Threshold Neural Network (ITNN) to enhance image clarity and perform segmentation through Segment Multiscale Neural Slicing (SMNS) to segment disease-affected regions. Further feature selection is done using Spectral Scaled Absolute Feature Selection (S2AFS) in combination with Social Spider Optimization with Closest Weight (S2O-FCW) to refine the most informative features. The classification is executed using a SoftMax activation-based DSGAN2 model trained to distinguish healthy from diseased samples. Experimental analysis, performed on a benchmark dataset demonstrates the model performance with better accuracy over existing models like CNN, AlexNet, and APS-DCCNN.

The feature selection framework reported in [18] incorporates an enhanced Salp Swarm Algorithm for plant disease detection. The presented methodology extracts handcrafted features from plant leaf images. SSAFS is used to identify the most informative subset of these features by simulating the foraging behavior of salps with a chaotic initialization strategy and Sine Cosine Algorithm-enhanced population evolution. The selected features are fed into a neural network classifier, and performance is validated across multiple UCI and PlantVillage datasets. Experimental results indicate that SSAFS outperforms standard optimization methods like PSO, ABC, and IBGWO in classification accuracy, feature reduction, and convergence speed. However, the model is limited by its reliance on handcrafted features and a lack of evaluation in real outdoor agricultural environments.

The ensemble deep learning framework presented in [19] is for the accurate recognition of cotton leaf diseases. The model combines a standard CNN with a fine-tuned VGG16 network, using transfer learning to enhance generalization from pre-trained ImageNet weights. The methodology involves preprocessing through data augmentation, followed by training individual networks that are later ensembled for improved classification robustness. The dataset includes six cotton disease categories and was built from both field-captured and PlantVillage images. Experimental analysis demonstrates superior accuracy over CNN and fine-tuned VGG models. However, limitations include class imbalance in the dataset and potential overfitting risks, particularly with rare diseases like Areolate and Myrothecium.

A similar ensemble-based deep learning approach is presented in [20] to detect plant leaf diseases incorporating DenseNet169, InceptionV3, and Xception models. The methodology focuses on combining the confidence scores from these pretrained CNNs through two custom-designed non-linear equations that emulate exponential and sigmoid behaviors to enhance decision reliability. This ensemble mechanism improves classification accuracy by minimizing deviation across confidence scores, effectively mitigating model bias. The experimental evaluation demonstrates the model peak accuracy over traditional ensemble techniques like fuzzy rank, soft voting, and weighted average. However, the model's high computational demands and limited interpretability of the fusion mechanism pose challenges for real-time deployment.

The hybrid approach presented in [21] for plant leaf disease classification combines multiple image processing and machine learning techniques. The presented methodology begins with resizing and contrast enhancement of tomato leaf images, followed by K-means clustering for segmentation and contour tracing for structural boundary detection. To extract meaningful features, it employs a combination of Discrete Wavelet Transform (DWT), Principal Component Analysis (PCA), and Gray-Level Co-occurrence Matrix (GLCM). These features are then classified using Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Convolutional Neural Network (CNN) classifiers. The results exhibit the CNN highest accuracy performance over other models. However, the model's limitation is present in its focus on handcrafted feature techniques. Also, its dependence on structured datasets and evaluation limited to leaf samples highlights the need for broader validation on diverse plant species and real-world images.

The deep learning model presented in [22] incorporates ResNet-50 and Inception-v3 as a hybrid model for early plant disease detection. The methodology includes structured preprocessing, data augmentation, and normalization to ensure robustness under real-world agricultural conditions. The hybrid model utilizes ResNet-50’s skip connections and Inception-v3’s multi-scale feature learning to provide better feature extraction. Experimental analysis demonstrated the model better classification performance with better accuracy, and F1-score. However, the results exhibit overfitting in validation trends for small epoch range which limits adaptability in real-time deployment.

The Hybrid Learning Model (HLM) presented in [23] combines Deep Reinforcement Learning (DRL) with Transfer Learning (TL) for early detection of plant diseases. The approach incorporates an advanced image preprocessing procedure. Adaptive Median Filter (AMF) and Color Histogram Techniques (CHT) are used to improve visual clarity and reduce noise. Further feature extraction is performed using the MobileNetV2 architecture, followed by fine-tuning done through reinforcement learning to classify diseases based on evolving decision patterns. The model was trained and tested on a benchmark dataset and the experimental results showed its outstanding performance over VGG19 and DoubleGAN models. However, the presented model faces challenges in computational efficiency due to DRL’s iterative training nature and lacks testing under diverse real-world agricultural conditions.

A visual information-guided multi-modal anomaly detection framework is presented in [24] for plant disease identification using vision-language models (VLMs). The model integrates visual guidance to optimize prompt tuning, enhancing anomaly detection by aligning visual features across known and unknown categories. Experimental evaluations on the PlantVillage dataset showed a significant improvement achieving high AUROC scores and reduced FPR outperforming baseline fine-tuned models. However, the presented approach remains limited by computational overhead and dependence on the quality of vision-language pretraining.

Research Gap: The literature review reveals several recurring limitations that justify the development of a novel, more robust plant disease classification framework. Many existing studies rely heavily on either shallow CNN architectures or traditional handcrafted feature extraction methods, limiting their ability to capture complex spatial hierarchies and inter-class visual similarities. Although a few models are lightweight and efficient, they often sacrifice fine-grained classification accuracy, especially in underrepresented or visually overlapping classes. Several approaches focus on limited disease categories or single-crop datasets, reducing generalizability to broader agricultural settings. Moreover, although some papers incorporate advanced optimizers or pretraining strategies, there is often a lack of effective feature fusion mechanisms to fully exploit multi-scale contextual information. Computational inefficiency, absence of real-time adaptability, and minimal attention to class imbalance further constrain performance in real-world deployments. Few models address integrated preprocessing, attention-guided learning, and hyperparameter tuning as a unified pipeline.

These gaps highlight the necessity for a more comprehensive solution that incorporates adaptive preprocessing, dual-path feature learning, attention-based fusion, and optimizer-driven parameter tuning. The proposed DPAFNet aims to overcome these limitations by providing a scalable, accurate, and generalizable deep learning architecture for complex, multi-class plant disease detection.

3. Proposed Work

The work introduces DPAFNet (Dual-Path Attention Fusion Network), a novel deep network for accurate and efficient banana plant disease classification. By merging MaxViT and HorNet-S, the architecture efficiently extracts both global and local feature representations. Using its hierarchical convolutions, HorNet-S develops local contextual awareness, whereas MaxViT offers multiscale attention mechanisms for learning semantic structures at higher levels. To achieve the best possible channel and spatial information fusion throughout the network, these two-stream outputs are then combined with a Cross-Layer Attention Fusion (CLAF) module. To manage an uneven distribution of data and provide improved generalization, a class-balanced focal loss function is employed. In addition, an adaptive quantum monarch butterfly optimization (AQMBO) algorithm is used to adapt hyperparameters in order to improve convergence through dynamic learning rate adjustment and weight regularization.

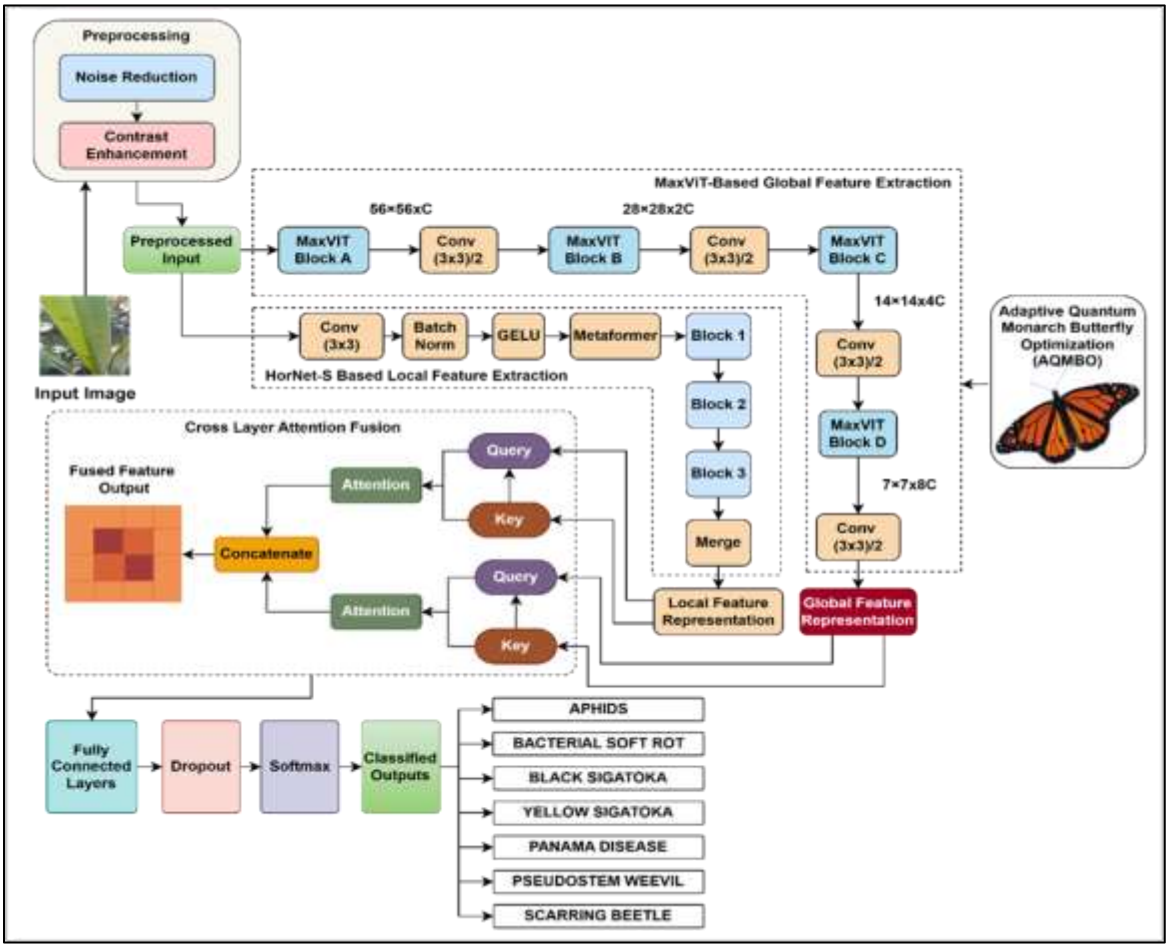


Figure 1. Proposed Model Overview

A learnable preprocessing step that refines raw input images is the first step in the process, as shown in Figure 1. The next step involves using the MaxViT and HorNet-S branches to extract features in parallel. After being combined using CLAF, these characteristics are sent to a classification head, which creates the final disease category. Because of this well-integrated pipeline, DPAFNet is a very dependable option for real-time plant disease diagnostics, improving both precision and recall.

3.1 Input Acquisition

Banana leaf photos are the first step in the diagnostic pipeline. These images are taken in a variety of environmental settings with handheld cameras, drones, or mobile devices. Since these photos frequently have changes in lighting, shadows, background noise, and clutter, the input acquisition step is crucial to guaranteeing the quality and interpretability of feature extraction procedures that follow. Let's assume that the raw image that was taken is represented mathematically as

$$\mathcal{I}_{r} \in R^{h\times w\times c} \tag{1}$$

where $\mathcal{I}_{r}$ indicates the raw image matrix, h indicates the height (in pixels), w indicates the width (in pixels), c indicates the number of channels (typically c = 3 for RGB color space). Due to outdoor conditions, the raw image is often affected by random fluctuations in illumination and sensor noise. To model the corrupted observation, we assume the presence of additive noise in the image acquisition process. The observed image $\mathcal{I}_{o}$ is mathematically formulated as

$$\mathcal{I}_{o}(x,y) = \mathcal{I}_{r}(x,y) + \epsilon(x,y) \tag{2}$$

Where $\mathcal{I}_{o}(x,y)$ indicates the observed pixel intensity at position (x,y), $\epsilon(x,y)$ indicates the noise term representing distortion or corruption at pixel (x,y). The noise component $\epsilon(x,y)$ is assumed to follow a Gaussian distribution, which is formulated as

$$\epsilon(x,y) \sim \mathcal{N}(0,\sigma^2) \tag{3}$$

where $\mathcal{N}(0,\sigma^2)$ indicates the normal distribution with mean zero and variance σ^2, σ^2 indicates the noise variance estimated from low-frequency background regions of the image. In practice, pixel values in $\mathcal{I}_{r}$ are restricted to the digital range [0,255] for 8-bit color channels. However, for compatibility with deep learning models, the observed image is rescaled to normalized floating-point values in the interval [0,1], using the transformation which is expressed as follows

$$\mathcal{I}_{n}(x,y,k) = \frac{\mathcal{I}_{o}(x,y,k)}{255} \tag{4}$$

where $\mathcal{I}_{n}(x,y,k)$ indicates the normalized intensity at pixel location (x,y) and channel k, $k \in \{1,2,3\}$ denotes the RGB channel index. Additionally, to ensure consistency across samples and compatibility with the model input dimensions, the image is resized to a fixed shape using bilinear interpolation which is mathematically expressed as

$$\mathcal{I}_{s} = \text{Resize}(\mathcal{I}_{n}, h_s, w_s) \tag{5}$$

where $\mathcal{I}_s$: size-adjusted image, h_s, w_s indicates the standardized height and width (e.g., 224×224 pixels). The final step in acquisition involves preparing the image for batch-based processing. A batch of N such images is stacked into a 4D tensor which is mathematically formulated as

$$\mathcal{B} = \{\mathcal{I}_s^{(1)}, \mathcal{I}_s^{(2)}, \dots, \mathcal{I}_s^{(\mathcal{N})}\} \in R^{N \times h_s \times w_s \times c} \qquad (6)$$

where $\mathcal{B}$ indicates the input batch tensor, $\mathcal{I}_s^{(i)}$ indicates the i^{th} image in the batch, * N: number of samples per batch. This batch $\mathcal{B}$ is then passed to the next stage of the model for preprocessing and feature extraction. The care taken in this stage ensures that noise is modeled correctly, dynamic range is normalized, and spatial resolution is consistent all of which are essential.

3.2 Learnable Preprocessing Module

After acquiring and normalizing the input batch $\mathcal{B} \in R^{N \times h_s \times w_s \times c}$, the images are passed through a learnable preprocessing module. Unlike conventional methods that apply static filters, this module is constructed using a shallow convolutional neural network designed to perform dynamic noise suppression and contrast enhancement, trained end-to-end with the rest of the network. The first stage of this module is a convolutional transformation which is mathematically expressed as

$$\mathcal{F}_1 = \sigma_1\big(\mathrm{BN}_1(W_1 * \mathcal{B} + b_1)\big) \qquad (7)$$

Where $\mathcal{F}_1 \in R^{N \times h_s \times w_s \times f_1}$ indicates the output feature map of the first layer, W_1 indicates the convolution kernel of shape $k_1 \times k_1 \times c \times f_1$, b_1 indicates the bias vector for the first layer, '*' indicates the 2D convolution, $\mathrm{BN}_1(\cdot)$ indicates the batch normalization, $\sigma_1(\cdot)$ indicates the activation function (e.g., ReLU or GELU), f_1 indicates the number of filters in layer 1, k_1: kernel size. The convolution step detects local patterns while also allowing for noise smoothing through learnable kernels. The batch normalization ensures stable gradient flow and compensates for internal covariate shift, while the activation introduces non-linearity to handle varying illumination. Next, a second layer deepens the transformation which is mathematically expressed as

$$\mathcal{F}_2 = \sigma_2\big(\mathrm{BN}_2(W_2 * \mathcal{F}_1 + b_2)\big) \qquad (8)$$

where $\mathcal{F}_2 \in R^{N \times h_s \times w_s \times f_2}$ indicates the refined feature map, W_2 indicates the kernel matrix of shape $k_2 \times k_2 \times f_1 \times f_2$, b_2 indicates the bias term of second layer, f_2 indicates the number of output channels, $\sigma_2(\cdot)$: activation function. This layer acts as an adaptive contrast enhancer, where the feature filters are trained to emphasize boundaries, lesions, and gradient regions that signify disease symptoms, even in the presence of varying image quality. A final convolutional layer is applied to transform the enhanced feature representation back to image-like structure which is mathematically formulated as

$$\mathcal{I}_p = \sigma_3(W_3 * \mathcal{F}_2 + b_3) \qquad (9)$$

where $\mathcal{I}_p \in R^{N \times h_s \times w_s \times c}$: preprocessed image batch (same shape as input), W_3 indicates the kernel for the final transformation with shape $k_2 \times k_2 \times f_1 \times f_2$, b_2 indicates the bias vector for the final layer, $\sigma_3(\cdot)$ indicates the activation function. The output $\mathcal{I}_p$ preserves the original image dimensions and is optimized to suppress noise, highlight discriminative regions, and normalize brightness and contrast. This learnable approach adapts to various acquisition environments by updating weights $\{W_1, W_2, W_3\}$ and biases $\{b_1, b_2, b_3\}$ during backpropagation. To summarize the full pipeline of the preprocessing module, it is defined as a composite function which is expressed as

$$\mathcal{I}_{\mathcal{p}} = \mathcal{G}(\mathcal{B}; \Theta_{pre}) \quad (10)$$

where $\mathcal{G}(\cdot)$: stacked operations, $\Theta_{pre} = \{W_1, W_2, W_3, b_1, b_2, b_3\}$ indicates the learnable parameters of the preprocessing module. This preprocessed batch $\mathcal{I}_{\mathcal{p}}$ becomes the input to the subsequent dual-path feature extraction stage, where both global and local disease traits are analyzed. The adaptability of the preprocessing ensures that feature extraction operates on images with enhanced quality, regardless of the initial acquisition conditions.

3.3 Dual-Path Feature Extraction

Following preprocessing, the enhanced image tensor $\mathcal{I}_{\mathcal{p}} \in R^{N \times h_s \times w_s \times c}$ is forwarded into a dual-path architecture designed to extract both global contextual features and local textural information. This two-stream design enhances the discriminative capability of the model across diseases with varying symptom patterns.

3.3.1 MaxViT-Based Global Feature Extraction

This stream captures long-range spatial dependencies and coarse-scale semantic structures using a hybrid CNN-transformer model. Initially, the preprocessed image is divided into n non-overlapping patches which are mathematically expressed as

$$P = \text{Split}(\mathcal{I}_{\mathcal{p}}), \quad P \in R^{N \times n \times p^2 \cdot c} \quad (11)$$

where P patch matrix, p patch dimension, $n = \frac{h_s \cdot w_s}{p^2}$ total number of patches. Each patch is passed through a linear projection to obtain a low-dimensional embedding which is formulated as

$$E = P \cdot W_e + b_e, \quad E \in R^{N \times n \times d} \quad (12)$$

where E embedded patch tokens, $W_e \in R^{(p^2 \cdot c) \times d}$ learnable weight matrix, $b_e \in R^d$ bias vector, d embedding dimension. The embedded sequence E is then input to a MaxViT block consisting of local convolutional encoding and two types of attention: block attention $\mathcal{A}_{\mathcal{b}}$ and grid attention $\mathcal{A}_{\mathcal{g}}$ which is formulated as

$$F_A = \mathcal{A}_{\mathcal{g}}\left(\mathcal{A}_{\mathcal{b}}(\mathcal{C}(E))\right) \quad (13)$$

where $F_A \in R^{N \times n \times d}$ indicates the output from MaxViT, $\mathcal{C}(\cdot)$ indicates the depthwise convolutional transformation, $\mathcal{A}_{\mathcal{b}}$ indicates the attention within local image blocks, $\mathcal{A}_{\mathcal{g}}$ indicates the attention across spatial grids. Each attention layer computes self-attention which is formulated as

$$\text{Attn}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}} + B\right)V \tag{14}$$

where $Q = EW_Q, K = EW_k, V = EW_V$ indicates the query, key, and value matrices, $W_Q, W_K, W_V \in R^{d \times d}$ indicates the trainable projections, d_k indicates the dimensionality of keys, B indicates the relative position bias. MaxViT's interleaved attention mechanism enables the model to integrate fine and coarse global information hierarchically. Figure 2 depicts an illustration of global feature extraction using MaxViT.

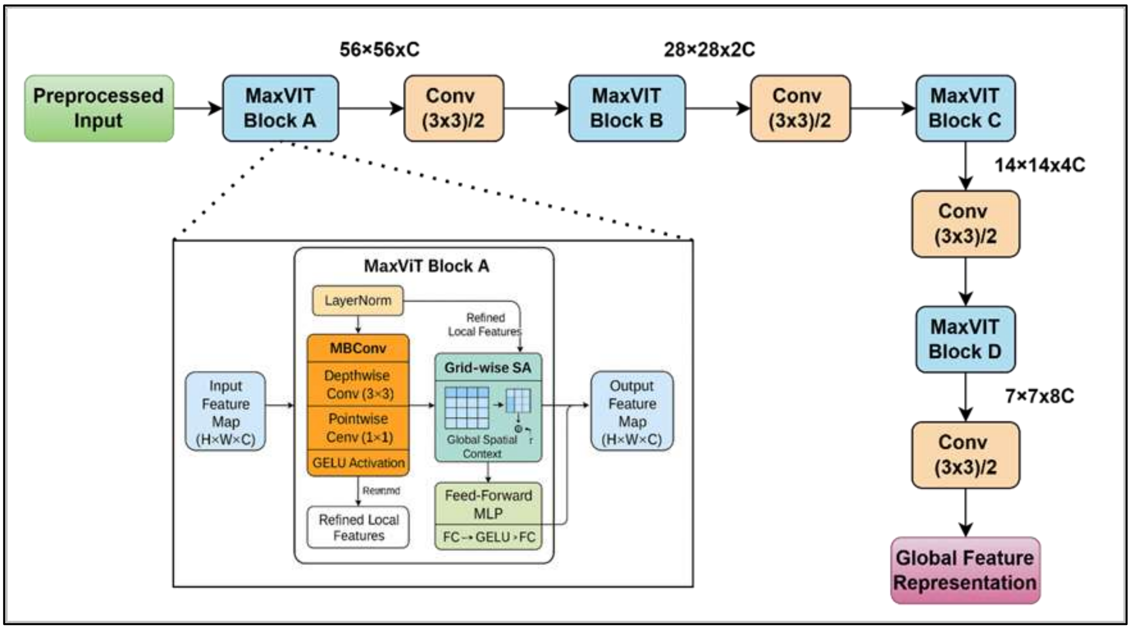


Figure 2. MaxViT-Based Global Feature Extraction

3.3.2 HorNet-S Based Local Feature Extraction

HorNet-S was selected over alternative CNN backbones due to its large-kernel convolutional design, which excels at modeling subtle textural variations in leaf surfaces. This property is especially important for visually similar diseases such as Yellow and Black Sigatoka. Additionally, HorNet-S maintains a lightweight structure, ensuring computational efficiency while preserving high discriminative power, making it an ideal complement to MaxViT's global feature extraction.

This stream is engineered to extract texture-level details, which are especially critical for fine-grained disease detection. The image $\mathcal{I}_{\wp}$ is processed through stacked large-kernel convolutions, formulated as

$$F_B^{(1)} = \phi\left(\text{BN}_3\left(W_4 * \mathcal{I}_{\wp} + b_4\right)\right) \tag{15}$$

$$F_B^{(l)} = \phi\left(\text{BN}_{l+2}\left(W_{l+3} * F_B^{(l-1)} + b_{l+3}\right)\right) \quad \text{for } l = 2, \ldots, L \tag{16}$$

where $F_B^{(l)} \in R^{N \times h_s \times w_s \times f_l}$ indicates the output of HorNet-S at layer l, $W_{l+3} \in R^{k \times k \times f_{l-1} \times f_l}$ indicates the convolution kernel with a large kernel size k, b_{l+3} indicates the bias term at layer l, $\phi(\cdot)$ indicates the GELU activation, L indicates the number of HorNet layers, f_l indicates the number of filters at layer l. The final output from this stream is formulated as

$$F_B = F_B^{(L)} \in R^{N\times h_s\times w_s\times f_L} \quad (17)$$

To align dimensionality with the MaxViT output F_A, a global average pooling (GAP) is applied which is mathematically expressed as

$$F_B' = \text{GAP}(F_B), \quad F_B' \in R^{N\times 1\times 1\times f_L} \quad (18)$$

It is then reshaped and linearly projected. Mathematically it is expressed as

$$F_B'' = F_B' \cdot W_t + b_t, \quad F_B'' \in R^{N\times n\times d} \quad (19)$$

where $W_t \in R^{f_L\times d}$ indicates the transformation matrix, $b_t \in R^d$ indicates the bias vector. This reshaping step ensures F_B'' matches the shape of F_A for downstream fusion. At the end of this stage, the model produces two parallel feature representations. $F_A \in R^{N\times n\times d}$ indicates the global context features from MaxViT, $F_B'' \in R^{N\times n\times d}$ indicates the fine-grained texture features from HorNet-S. These are passed to the Cross-Layer Attention Fusion module, where they are merged to form a unified representation optimized for robust classification across banana disease types. Figure 3 depicts an illustration of local feature extraction using HorNet-S.

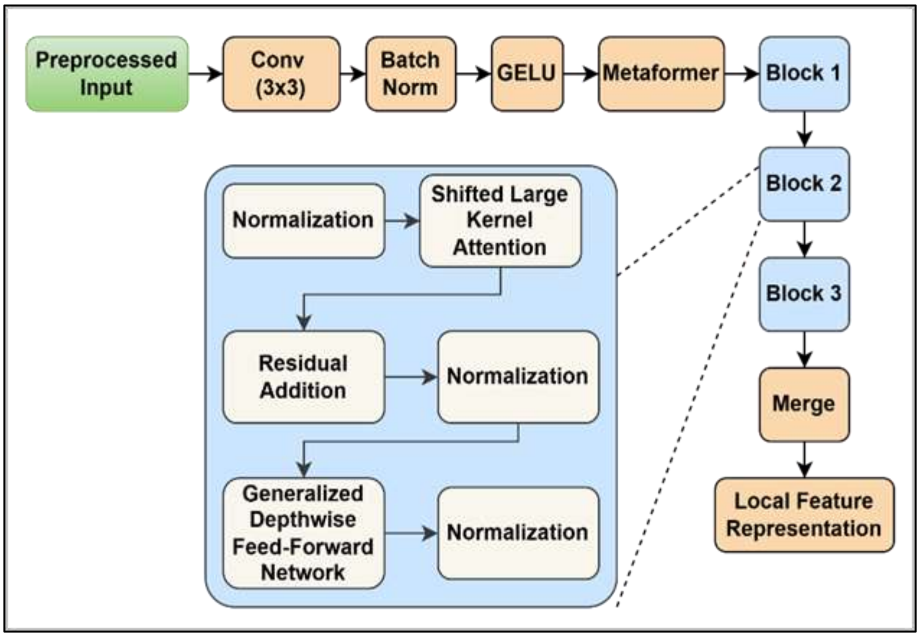


Figure 3. HorNet-S Based Local Feature Extraction

Unlike a single-path encoder–decoder that sequentially processes feature, the proposed dual-path architecture enriches feature representations by combining MaxViT's global context learning with HorNet-S's fine-grained texture extraction. This complementary fusion ensures that both long-range dependencies and subtle lesion patterns are preserved, thereby enhancing the discriminative power for visually overlapping banana disease categories. The dual-path attention fusion architecture improves feature richness because it extracts and merges complementary information streams that a single-path encoder–decoder cannot capture. In the proposed DPAFNet, one path uses MaxViT to capture long-range spatial dependencies and coarse semantic structures, while the other path uses HorNet-S to capture fine-grained textural details critical for distinguishing visually similar banana diseases. These two parallel feature representations are then combined through the Cross-Layer Attention Fusion (CLAF) module, which applies both channel and spatial attention before residual merging. This ensures that disease-relevant patterns are selectively emphasized, enhancing the discriminative capability of the network across diverse and overlapping disease symptoms.

Compared to a single-path encoder–decoder, which processes features sequentially and often misses either global dependencies or local textures, the dual-path design enriches the representation space by preserving and fusing both hierarchical context and detailed lesion morphology. This leads to superior performance, as reflected in the consistent improvements in precision, recall, and class-wise F1 scores reported in the experimental results.

3.4 Cross-Layer Attention Fusion (CLAF)

After extracting features from two distinct paths such as global context features $F_A \in R^{N\times n\times d}$ from MaxViT and local detail features $F_B'' \in R^{N\times n\times d}$ from HorNet-S. The next objective is to merge these representations in a meaningful and learnable manner. The Cross-Layer Attention Fusion (CLAF) module is designed to emphasize disease-relevant patterns by utilizing channel and spatial attention mechanisms, followed by a residual combination. Both streams already yield feature tensors of equal dimensions. For clarity, it is represented as

$$F_G = F_A, \quad F_L = F_B'' \tag{20}$$

where F_G indicates the global token matrix from Path A, F_L indicates the local token matrix from Path B, N indicates the batch size, n indicates the number of patches, d indicates the embedding size. These tensors are the input to parallel attention refinement branches.

The fusion layer resolves potential feature map conflicts and redundancy between the MaxViT and HorNet-S branches through the Cross-Layer Attention Fusion (CLAF) mechanism. This module applies channel-wise and spatial attention to selectively reweight overlapping activations, ensuring that disease-relevant patterns are emphasized while redundant signals are suppressed. The residual combination preserves complementary information from both branches, allowing the model to retain diverse contextual and textural cues without feature dilution. This learnable fusion strategy effectively integrates global and local representations into a unified feature map optimized for classification.

To extract channel-wise dependencies from global features, channel attention masks are mathematically formulated as

$$M_C = \sigma\left(W_2^C \cdot \delta\left(W_1^C \cdot \mathrm{GAP}(F_G)\right)\right) \tag{21}$$

$$F_G^C = F_G \odot M_C \tag{22}$$

where $M_C \in R^{N\times 1\times d}$ indicates the channel attention mask, GAP$(\cdot)$ indicates the global average pooling over patches, $W_1^C \in R^{d\times d_r}, W_2^C \in R^{d_r\times d}$ are the projection matrices, $\delta(\cdot)$ indicates ReLU activation, $\sigma(\cdot)$ indicates the sigmoid function, $\odot$ indicates element-wise multiplication, d_r indicates the reduction dimension. This highlights key semantic dimensions by amplifying relevant channels and suppressing redundant ones. To capture spatially significant regions, attention across patches is computed from local features. Mathematically it is expressed as

$$M_S = \sigma(Conv_{1\times 1}[AvgP(F_L); MaxP(F_L)]) \quad (23)$$

$$F_L^S = F_L \odot M_S \quad (24)$$

where $M_S \in R^{N\times n\times \mathbb{1}}$ indicates the spatial attention mask, $AvgP, MaxP$ indicates average and max pooling across channels, $Conv_{1\times 1}$ indicates the pointwise convolution, $\sigma(\cdot)$ indicates sigmoid activation. This step localizes important spatial patches, enabling sharper boundary and lesion focus. After attention refinement, the outputs F_G^C and F_L^S are fused. The fusion incorporates a learnable scalar gate $\lambda \in [0,1]$ to balance contributions. Mathematically it is expressed as

$$F_{\text{fused}} = \lambda \cdot F_G^C + (1-\lambda)\cdot F_L^S + \eta \cdot (F_G - F_L) \quad (25)$$

where $F_{\text{fused}} \in R^{N\times n\times d}$ indicates fused representation, λ indicates trainable attention gate, η indicates residual scaling factor. The last term introduces a residual discrepancy compensation, encouraging the model to learn the difference between global and local cues. The fused tensor is projected to match the classifier's input requirements. A normalization and projection layer are applied which is formulated as follows

$$F_{\text{final}} = \text{LN}(F_{\text{fused}}) \cdot W_f + b_f \quad (26)$$

where $LN(\cdot)$ indicates layer normalization, $W_f \in R^{d\times d}$ indicates the final projection matrix, $b_f \in R^d$ indicates the projection bias. $F_{\text{final}} \in R^{N\times n\times d}$ is the output passed to classification head. The Cross-Layer Attention Fusion module ensures that global semantics and local textures are adaptively merged. It learns not only to enhance but also to weigh the importance of each feature pathway dynamically, strengthening the classifier's robustness to subtle or overlapping disease symptoms.

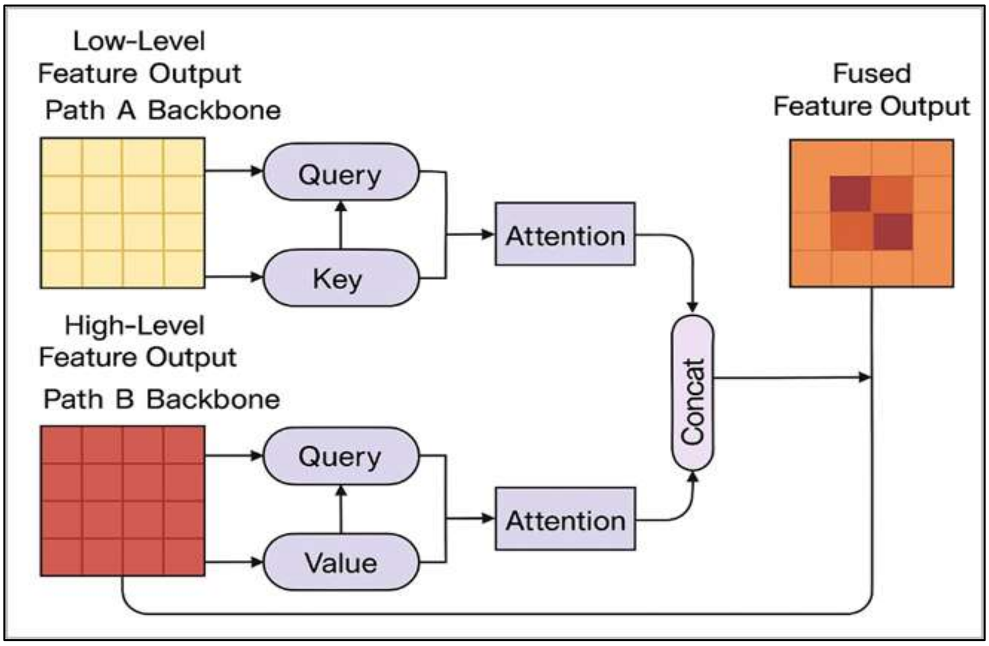


Figure 4. Cross Layer Attention Fusion

3.5 Classification Head

Once the fused feature representation $F_{\text{final}} \in R^{N\times n\times d}$ is obtained from the Cross-Layer Attention Fusion (CLAF) module, it is passed into the classification head, which is responsible for producing the final disease category prediction for each input image. This stage involves token aggregation, fully connected transformation, and SoftMax-based probability estimation. To consolidate the patch-wise information into a compact vector per image, a token aggregation operation is applied across the patch dimension which is mathematically expressed as

$$v_i = \text{Mean}\left(F_{\text{final}}^{(i)}\right), \quad v_i \in R^d \tag{27}$$

where $F_{\text{final}}^{(i)} \in R^{n\times d}$ indicates the feature matrix for the i-th image, $Mean(\cdot)$ indicates the average across all patches, v_i indicates the aggregated embedding vector for image i, d indicates the feature dimension. After processing all N images, the result is a matrix which is given as follows

$$V = [v_1, v_2, \dots, v_N]^T \in R^{N\times d} \tag{28}$$

This vector V summarizes all contextual and spatial features for each input in a single descriptor. The aggregated feature vector V is passed through a linear classifier, which maps it to a set of class scores as follows

$$z_j^{(i)} = V^{(i)} \cdot w_j + b_j \quad \text{for } j = 1,2, \dots, C \tag{29}$$

where $z_j^{(i)}$ indicates the score assigned to class j for sample i, $w_j \in R^d$: weight vector for class j, $b_j \in R$: bias for class j, C: total number of classes, $V^{(i)}$ indicates the aggregated feature vector for sample i. Stacking all outputs produces the score matrix which is mathematically expressed as

$$Z = V \cdot W_c + B_c \in R^{N\times C} \tag{30}$$

where $W_c \in R^{d\times C}$ indicates the classification weight matrix, $B_c \in R^{1\times C}$ indicates the bias vector for all classes, Z indicates the unnormalized logits for the batch. The raw scores in Z are converted into class probabilities using the SoftMax function which is mathematically expressed as

$$\hat{y}_j^{(i)} = \frac{e^{z_j^{(i)}}}{\sum_{k=1}^{C} e^{z_k^{(i)}}} \tag{31}$$

where $\hat{y}_j^{(i)} \in [0,1]$ is the predicted probability of sample i belonging to class j, $\sum_{j=1}^{C} \hat{y}_j^{(i)} = 1$ ensuring the output forms a valid probability distribution. The resulting matrix $\hat{Y} \in R^{N\times C}$ contains the predicted class probabilities for each image in the batch. The predicted class label $\hat{c}_i$ for sample i is determined by selecting the class with the highest predicted probability. Mathematically it is expressed as

$$\hat{c}_i = \arg\max_j \hat{y}_j^{(i)} \tag{32}$$

where $\hat{c}_i \in \{1,2,\ldots,C\}$ is the final class prediction for input i. This result serves as the output of the model and indicates which banana disease (or healthy status) is identified for each test image. The output of the classification head is a matrix $\hat{Y} \in R^{N\times C}$, where each row corresponds to the predicted class probability vector for an image. The training objective is to minimize a loss function that not only penalizes incorrect predictions but also accounts for class imbalance and difficult samples. For this purpose, a Class-Balanced Focal Loss is used, which combines two components: class frequency reweighting and focal modulation. Let the ground truth label for sample i be encoded as follows

$$y_i = \left[y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(C)}\right] \in \{0,1\}^C \tag{33}$$

where $y_i^{(j)} = 1$ if image i belongs to class j, otherwise 0, C indicates the total number of classes, N indicates the number of training samples in a batch. To compute Class Frequency and Inverse Weighting, consider the number of samples belonging to class j be f_j. The class weight α_j is formulated as

$$\alpha_j = \frac{1}{\log(1+f_j)} \tag{34}$$

where $\alpha_j \in R$ indicates the weight assigned to class j, inversely proportional to its frequency, f_j indicates the total count of class j in the dataset. This ensures minority classes contribute more to the loss, while frequent classes are down-weighted. To emphasize hard-to-classify samples and reduce the impact of well-classified ones, a modulating factor is applied to each predicted probability which is mathematically expressed as

$$m_i^{(j)} = \left(1 - \hat{y}_i^{(j)}\right)^{\gamma} \tag{35}$$

where $\hat{y}_i^{(j)}$ indicates the predicted probability for class j on image i, $m_i^{(j)}$ indicates the modulation weight, $\gamma \in R_+$ indicates the focusing parameter. A higher γ places more weight on misclassified examples. To compute Class-Balanced Focal Loss per Sample the loss for image i with respect to class j is formulated as

$$\mathrm{l}_i^{(j)} = -\alpha_j \cdot m_i^{(j)} \cdot y_i^{(j)} \cdot \log\left(\hat{y}_i^{(j)}\right) \tag{36}$$

where $\mathrm{l}_i^{(j)}$ indicates the individual cross-entropy-based loss for class j on sample i, α_j indicates the class-balancing coefficient, $m_i^{(j)}$ indicates the focal modulation, $y_i^{(j)}$ indicates the binary ground truth indicator, $\log\left(\hat{y}_i^{(j)}\right)$ indicates the logarithmic penalty for predicted confidence. To aggregate total loss across batch and classes, the complete loss for a mini-batch of N samples is formulated as

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} \mathrm{l}_i^{(j)} \tag{37}$$

substituting $\mathrm{l}_i^{(j)}$

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} \alpha_j \cdot \left(1 - \hat{y}_i^{(j)}\right)^{\gamma} \cdot y_i^{(j)} \cdot \log\left(\hat{y}_i^{(j)}\right) \tag{38}$$

where $\mathcal{L}$ indicates the overall batch-averaged class-balanced focal loss, N indicates the batch size, C indicates the number of classes. This loss is minimized during training using gradient-based optimization.

3.6 Adaptive Quantum Monarch Butterfly Optimization (AQMBO)

The performance of deep neural networks like DPAFNet significantly depends on the choice of hyperparameters such as learning rate, dropout rate, attention depth, and regularization strength. Manual tuning is inefficient and prone to suboptimal results. To address this, the Adaptive Quantum Monarch Butterfly Optimization (AQMBO) algorithm is employed.

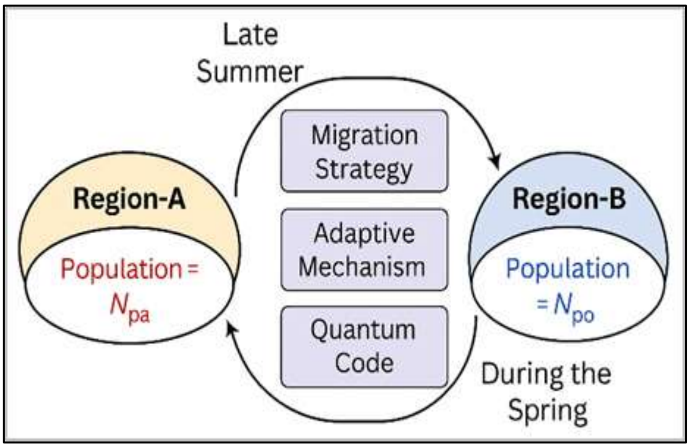


Figure 5. Adaptive Quantum Monarch Butterfly Optimization

It combines the migration behavior of monarch butterflies, quantum-inspired randomness, and adaptive update control for efficient convergence in high-dimensional search spaces. To define the hyperparameter search space the hyperparameter vector is denoted as

$$\theta_i^t = [\eta_i^t, \lambda_i^t, d_i^t, p_i^t, \alpha_i^t] \tag{39}$$

where θ_i^t indicates the hyperparameter vector of the i^{th} butterfly at iteration t, η_i^t indicates the learning rate, λ_i^t indicates the weight decay, d_i^t indicates the attention depth, p_i^t indicates the dropout rate, α_i^t indicates the fusion gate weight. The algorithm optimizes θ_i^t to minimize the loss $\mathcal{L}(\theta_i^t)$, evaluated via model validation. Let the population consist of P candidate butterflies $\Theta^0 = \{\theta_1^0, \theta_2^0, \dots, \theta_P^0\}$, Each θ_i^0 is randomly sampled within its allowed range $\theta_i^0(j) \sim \mathcal{U}(\theta_{\min}(j), \theta_{\max}(j))$ in which $\theta_i^0(j)$ indicates the j^{th} component of vector θ_i^0, $\mathcal{U}(\cdot)$: uniform distribution over feasible bounds. Further the fitness function is defined as the validation loss

$$f_i^t = \mathcal{L}_{\text{val}}(\theta_i^t) \tag{40}$$

where f_i^t indicates the fitness value of the butterfly i at iteration t. Lower f_i^t implies better hyperparameter performance. The global best solution at iteration t is expressed as

$$\theta_g^t = \arg\min_{\theta_i^t \in \Theta^t} f_i^t \tag{41}$$

Each butterfly updates its position based on its proximity to the global best which is mathematically expressed as

$$\theta_i^{t+1} = \theta_i^t + r \cdot \left(\theta_g^t - \theta_i^t\right) \tag{42}$$

where $r \sim \mathcal{U}(0, \beta_t)$: adaptive step size, β_t: migration coefficient that decreases with t, defined as

$$\beta_t = \beta_0 \cdot \left(1 - \frac{t}{T}\right) \tag{43}$$

β_0 indicates the initial migration influence, T indicates the total number of iterations. This mechanism promotes exploration in early iterations and exploitation in later phases. To escape local optima, quantum perturbation is introduced with probability δ

$$\theta_i^{t+1} = \theta_i^{t+1} + \epsilon_q \cdot sin(\phi) + \epsilon_q' \cdot cos(\phi) \tag{44}$$

where $\phi \sim \mathcal{U}(0,2\pi)$ random angle, ϵ_q, ϵ_q' small step size constants, the sine and cosine terms introduce multi-directional randomness, this mimics quantum tunneling across barriers in the loss landscape. The new position θ_i^{t+1} is accepted if it leads to improved performance:

$$f_i^{t+1} < f_i^t \Rightarrow \text{accept } \theta_i^{t+1}, \text{ else retain } \theta_i^t \tag{45}$$

This elitist selection ensures the algorithm does not regress in fitness. The algorithm continues until a stopping criterion is met $t = T$ or no improvement for τ generations. At termination, the optimal hyperparameter vector is

$$\theta^* = \theta_g^T \tag{46}$$

This optimized vector θ^* is then used to train the final DPAFNet model. The AQMBO algorithm ensures effective navigation through the hyperparameter space, balancing between exploration and convergence. The incorporation of adaptive migration and quantum-inspired randomness allows the model to reach globally optimal configurations for training, thereby improving classification accuracy, convergence speed, and generalization performance across diverse banana disease classes.

Pseudocode for the proposed DPAFNet with AQMBO for Banana Plant Disease Classification
Input: Raw image set $\mathcal{D} = \{\mathcal{I}_r^1, \mathcal{I}_r^2, \dots, \mathcal{I}_r^{\mathcal{M}}\}$, *Labels* $\mathcal{Y} = \{y^1, y^2, \dots, y^M, where y^i \in \{1,2, \dots, C\}$
Output: Predicted class $\hat{y}^i$ *for each image, Optimized model parameters* Θ^*
Begin *Initialization: Set maximum iterations* T, *population size* P, *quantum rate* δ, *Define hyperparameter bounds for* $\theta = [\eta, \lambda, d, p, \alpha]$, *Initialize butterfly population* $\Theta^0 = \{\theta_1^0, \theta_2^0, \dots, \theta_P^0\}$, *Set* $t = 0$ *For each* $\mathcal{I}_r^i \in \mathcal{D}$ *Normalize pixel values to* $[0,1]$ *Resize to fixed dimensions* $h \times w$ *Stack to form batch tensor* $\mathcal{B}$

For each image in $\mathcal{B}$
Apply three-layer CNN
$\mathcal{F}_1 = \phi\big(BN(W_1 * \mathcal{I} + b_1)\big),\ \mathcal{F}_2 = \phi\big(BN(W_2 * \mathcal{F}_1 + b_2)\big),\ \mathcal{I}_{\wp} = \phi(W_3 * \mathcal{F}_2 + b_3)$
Initialize Dual-Path Feature Extraction
For each $\mathcal{I}_{\wp}$
Global Path: Split into patches, embed with $E = P \cdot W_e + b_e$
Apply MaxViT $F_A = \mathcal{A}_g\left(\mathcal{A}_b\big(\mathcal{C}(E)\big)\right)$
Local Path
Pass through HorNet-S $F_B^{(1)} = \phi\left(BN\big(W_4 * \mathcal{I}_{\wp} + b_4\big)\right)$
Repeat L times $\rightarrow F_B^{(L)} = F_B$
Global pool and transform $F_B'' = GAP(F_B) \cdot W_t + b_t$
Cross-Layer Attention Fusion (CLAF)
Input: F_A, F_B''
Compute channel attention $M_C = \sigma\left(W_2^C \cdot \delta\Big(W_1^C \cdot GAP(F_A)\Big)\right), F_A^C = F_A \odot M_C$
Compute spatial attention $M_S = \sigma\big(Conv_{1x1}([AvgP(F_B''); MaxP(F_B'')])\big), F_B^S = F_B'' \odot M_S$
Fuse with residual $F_{fused} = \lambda \cdot F_A^C + (1-\lambda) \cdot F_B^S + \eta \cdot (F_A - F_B'')$
Normalize and project $F_{final} = LN\big(F_{fused}\big) \cdot W_f + b_f$
Initialize Classification Head
Aggregate patches $V_i = Mean\Big(F_{final}^{(i)}\Big)$
Compute scores $Z = V \cdot W_c + B_c$
Convert to probabilities $\widehat{Y}_i = SoftMax(Z_i)$
Predict $\hat{y}^i = arg\max_j \left(\widehat{Y}_i^{(j)}\right)$
For each sample i*, class* j
Obtain $\alpha_j = 1/\log\big(1 + f_j\big)$
$m_i^{(j)} = \left(1 - \widehat{Y}_i^{(j)}\right)^{\gamma}$
$\mathcal{L}_i^{(j)} = -\alpha_j \cdot m_i^{(j)} \cdot y_i^{(j)} \cdot \log\left(\widehat{Y}_i^{(j)}\right)$
Compute total loss $\mathcal{L} = \frac{1}{N}\sum_i \sum_j l_i^{(j)}$
AQMBO Optimization Loop
While $t < T$
For each butterfly $\theta_i^t \in \Theta^t$
Train model using θ_i^t
Compute fitness $f_i^t = \mathcal{L}_{val}(\theta_i^t)$
Find best $\theta_g^t = arg\min f_i^t$
For each butterfly
Update $\beta_t = \beta_0(1 - t/T), \theta_i^{t+1} = \theta_i^t + r \cdot \left(\theta_g^t - \theta_i^t\right)$
Apply quantum update with probability δ

$\theta_i^{t+1} += \epsilon_q \cdot sin(\phi) + \epsilon_q' \cdot cos(\phi)$

Accept if improved

$f_i^{t+1} < f_i^t \Rightarrow \theta_i^t = \theta_i^{t+1}$

Increment $t = t + 1$

Final Training Use $\theta^* = \theta_g^T$

Train final DPAFNet with optimal configuration

Deploy for testing and deployment

End

4. Results and Discussion

The proposed DPAFNet framework was evaluated on a benchmark banana leaf disease dataset from the Mendeley repository. The dataset comprises a balanced mix of healthy and diseased samples. The preprocessing stage includes an adaptive filtering module that dynamically enhances image contrast and suppressed noise prior to feature extraction. The dual-path feature extraction unit, integrating MaxViT and HorNet-S networks, captures both high-level spatial semantics and low-level texture information. Fusion was accomplished through a Cross-Layer Attention Fusion (CLAF) module, followed by classification using a fully connected SoftMax head. To enhance generalization and convergence, the Adaptive Quantum Monarch Butterfly Optimization (AQMBO) algorithm was applied for hyperparameter tuning. The model was trained and tested using an 80:20 ratio. Experiments were conducted in an NVIDIA GPU-accelerated environment with PyTorch 2.1 and Python 3.11. The optimized hyperparameters listed in Table 1 are the values selected by AQMBO after 50 search iterations.

Table 1. Simulation Hyperparameters of Proposed Model

S.No	Parameter	Value
1	Input Image Size	224 × 224 pixels
2	Number of Epochs	100
3	Batch Size	32
4	Optimizer	AdamW
5	Initial Learning Rate	0.0002
6	Learning Rate Scheduler	Cosine Annealing
7	Dropout Rate	0.25
8	Backbone Networks	MaxViT (Global), HorNet-S (Local)
9	Feature Fusion Module	Cross-Layer Attention Fusion (CLAF)
10	Optimizer Enhancer	AQMBO
11	Weight Decay	0.0001
12	Gradient Clipping	Enabled (Max norm: 5.0)
13	Attention Depth	6

14	Activation Function	GELU (Gaussian Error Linear Unit)

The dataset utilized in this study involves a diverse spectrum of banana leaf diseases and pest-related conditions, serving as the foundational component for training and validating the proposed DPAFNet model. Collected from the publicly available Mendeley Data Repository [25], the dataset includes high-resolution images grouped into seven well-defined classes: Aphids, Bacterial Soft Rot, Black Sigatoka, Panama Disease, Pseudostem Weevil, Scarring Beetle, and Yellow Sigatoka. Each category corresponds to a distinct pathological or pest-induced anomaly, making the dataset highly relevant for real-world agricultural diagnostics. The details about the dataset samples are presented in Table 2.

Table 2. Dataset Description

S.No	Class Name	Train Samples	Test Samples	Total Samples
1	Aphids	246	120	366
2	Bacterial Soft Rot	753	325	1,078
3	Black Sigatoka	337	137	474
4	Panama Disease	62	40	102
5	Pseudostem Weevil	1,928	808	2,736
6	Scarring Beetle	105	45	150
7	Yellow Sigatoka	188	76	264
	Total	**4,421**	**749**	**5,170**

The evaluation metrics used here are accordingly defined as

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{47}$$

$$Precision = \frac{TP}{TP+FP} \tag{48}$$

$$Recall = \frac{TP}{TP+FN} \tag{49}$$

$$F1 = \frac{Precision.Recall}{Precision+Recall} \tag{50}$$

Figure 6 shows the results obtained after the preprocessing part of the proposed DPAFNet model. The outcomes describe a significant improvement in the image and the capability of visualizing the region of the disease that appears in various categories of banana disease. The configured learnable preprocessing block adapts the intensities of pixels and spatial designs according to the relevance of features, so that input images are optimized for further learning. Using adaptive filtering, non-relevant background noise is effectively overcome while important structural attributes like lesion edges, discoloration, and fungal textures are simultaneously maintained and enhanced. The module also carries out intelligent contrast enhancement, where subtle changes in areas affected by the infection that might not be visible with raw imagery are magnified. This preprocessing procedure not only advances visual clarity but also allows feature extractors to concentrate on the most informative areas.

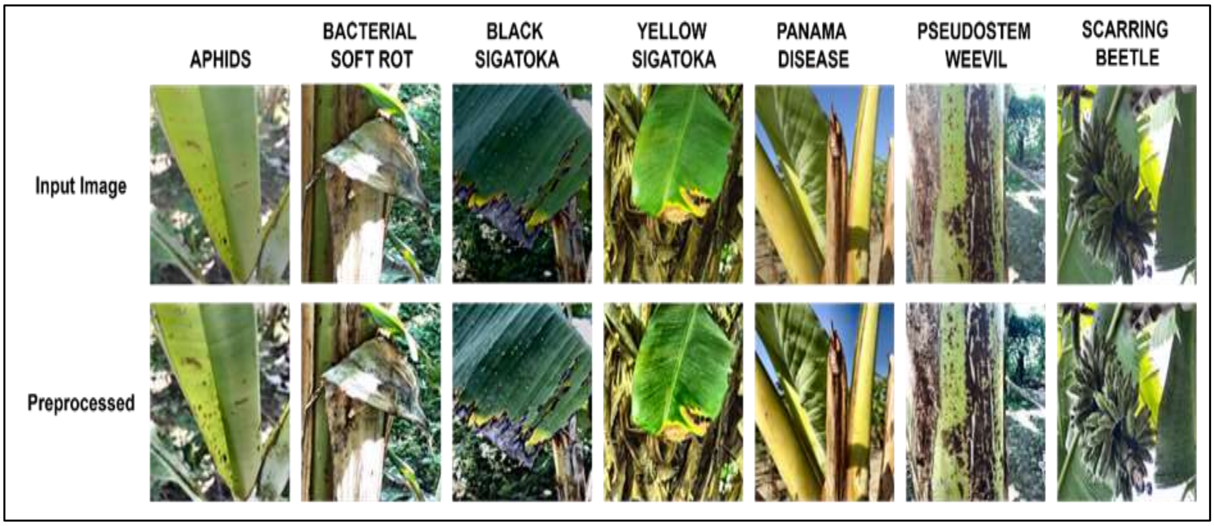


Figure 6. Input Samples and the Preprocessing Outputs

The accuracy of the proposed DPAFNet model on the train and validation set, represented in Figure 7, shows a high improvement trend throughout the entire 100 training epochs. Starting with an initial accuracy of about 9.6%, the model returns to more than 94.5% in training after the first epoch and is maintained at 98.3% in validation by the last epoch. Such improvement in the first 20 epochs represents an effective representation of learning achieved by the dual-path architecture that combines the MaxViT and HorNet-S feature extractors. The similarity between training and validation curves during the learning process ensures a high level of generalization which is additionally supported by the class-balanced focal loss and adaptive tuning through the AQMBO algorithm.

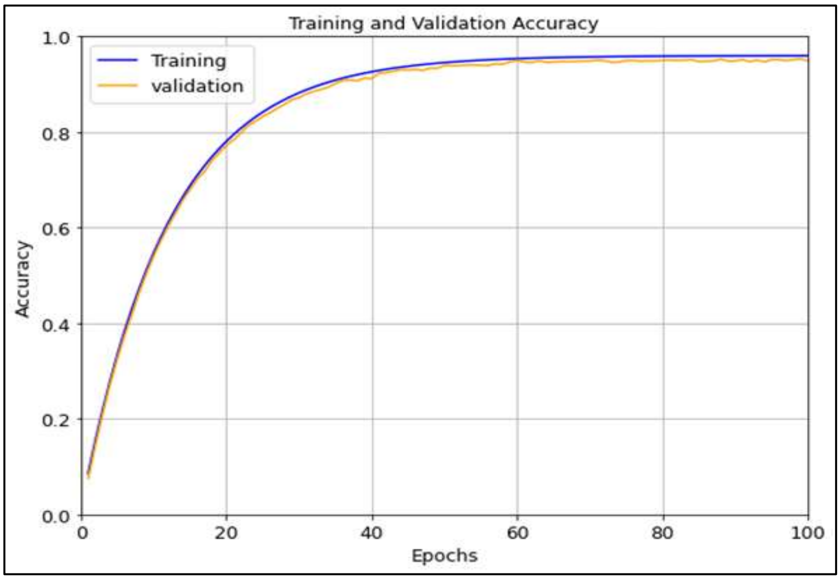


Figure 7. Training and Validation Accuracy of Proposed Model

There is a corresponding decrease in loss over time, as shown in Figure 8, with the training and validation loss curves plummeting to about 0.06 and 0.08, respectively, starting above the 0.9 bracket in the first years. Convergence is smooth and parallel, indicating that the model is learning and not degrading as the process progresses. The decline in the first 2530 epochs correlates with faster convergence on simpler class boundaries, while the smaller decline thereafter indicates refinement in the precision of harder classes of diseases. The small difference between the two curves demonstrates the successful regularization and generalization of the architecture. The combination of these trends confirms the robustness of the proposed DPAFNet in terms of both high performance and stable learning dynamics on difficult classification tasks.

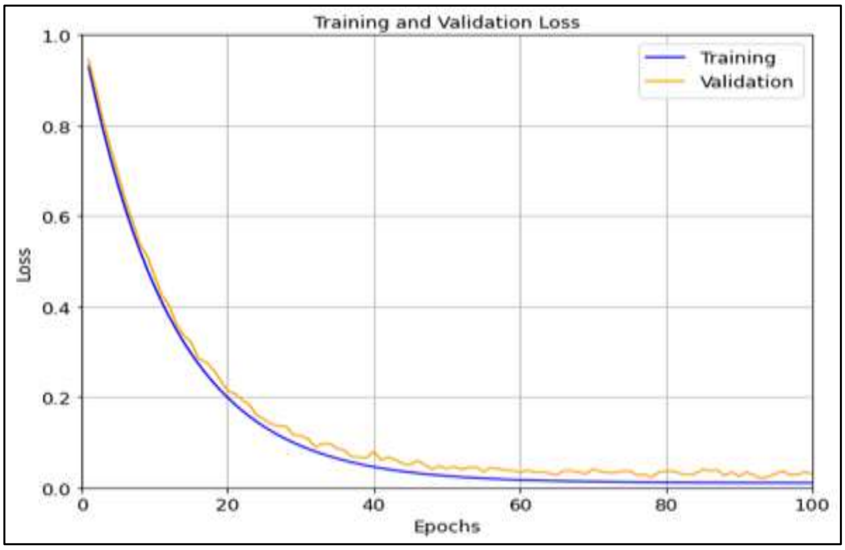


Figure 8. Training and Validation Loss of Proposed Model

The Precision-Recall (PR) curves shown in Figure 9 and 10 display the class-specific ability to detect using the proposed DPAFNet model during the training and testing periods. As evident in the raining PR curve (Figure 9), the model has perfect average precision (AP) scores in all seven classes thus learning is highly effective.

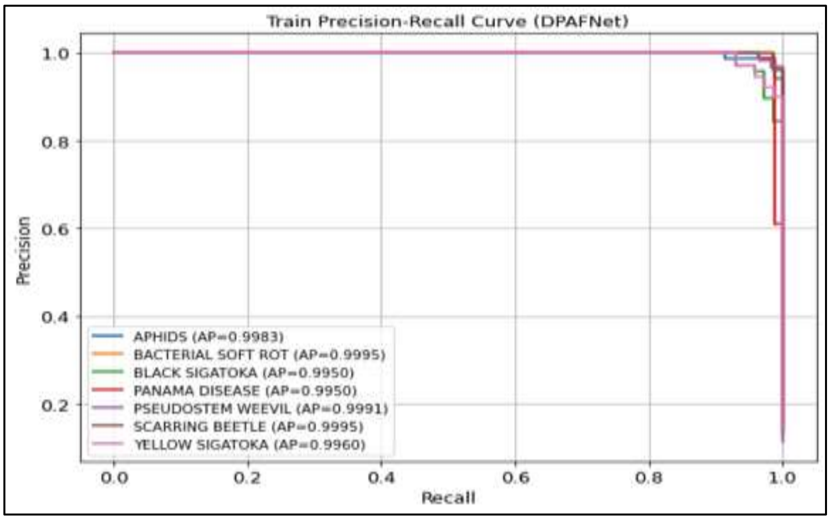


Figure 9. Precision-Recall Analysis of Proposed Model Training

Notably, Calibrations, such as Scarring Beetle, Pseudostem Weevil, and Bacterial Soft Rot reach AP values as high as 0.9995 with extremely accurate precision and recall. Agents such as Panama Disease and Black Sigatoka, which have high levels of complexity, had APs of 0.9950, confirming the high performance of the dual-path feature fusion and dynamic optimization approach. The DPAFNet remains highly effective during the testing phase, as the

APs of Aphids and Black Sigatoka remain at 0.9981 and 0.9935, respectively, in Figure 10. Minor decreases are seen in Bacterial Soft Rot (AP=0.9785) and Panama Disease (AP=0.9883), as expected because of the class imbalance and because the feature richness is shared.

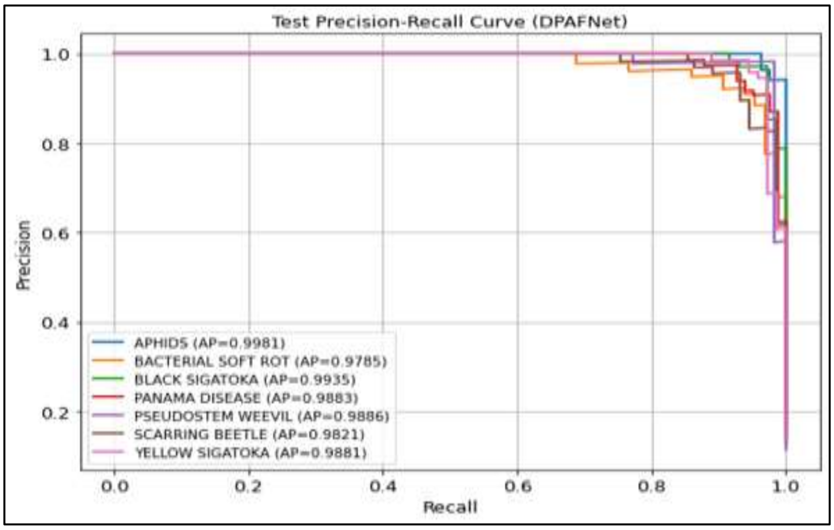


Figure 10. Precision-Recall Analysis of Proposed Model Testing

There is a corresponding decrease in loss over time, as shown in Figure 8, with the training and validation loss curves plummeting to about 0.06 and 0.08, respectively, starting above the 0.9 bracket in the first years. Convergence is smooth and parallel, indicating that the model is learning and not degrading as the process progresses. The decline in the first 2530 epochs correlates with faster convergence on simpler class boundaries, while the smaller decline thereafter indicates refinement in the precision of harder classes of diseases.

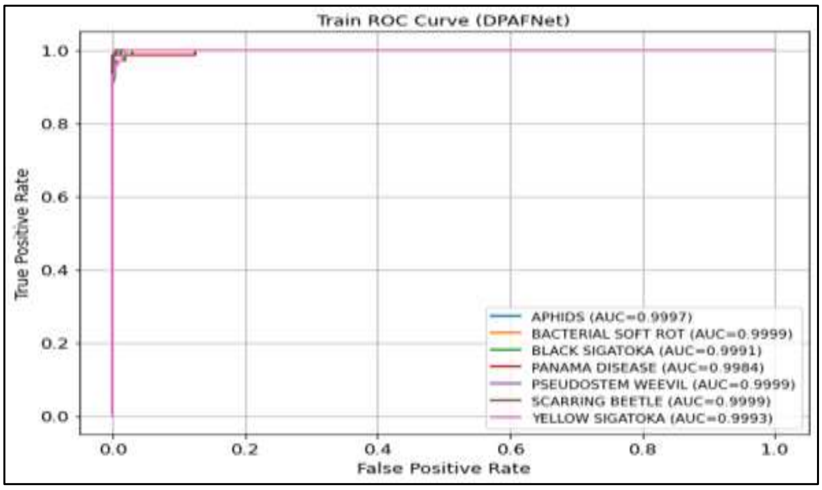


Figure 11. ROC Analysis of Proposed Model Training

The small difference between the two curves demonstrates the successful regularization and generalization of the architecture. The combination of these trends confirms the robustness of the proposed DPAFNet in terms of both high performance and stable learning dynamics on difficult classification tasks.

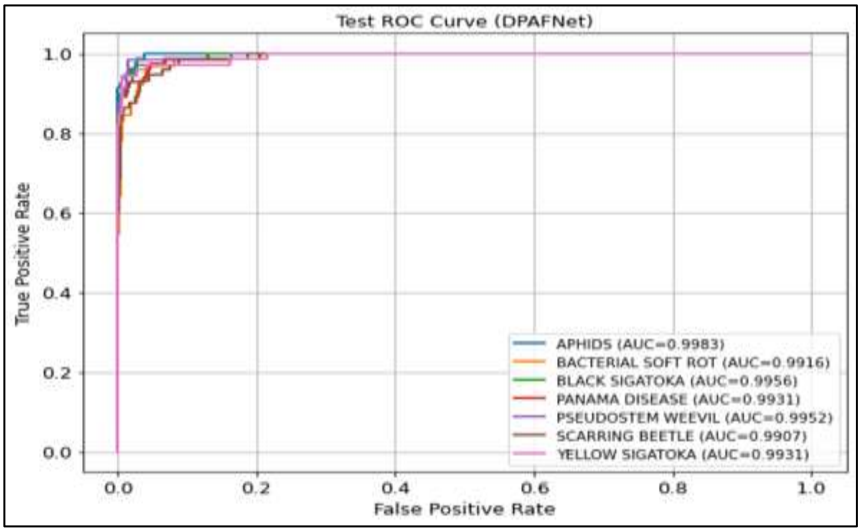
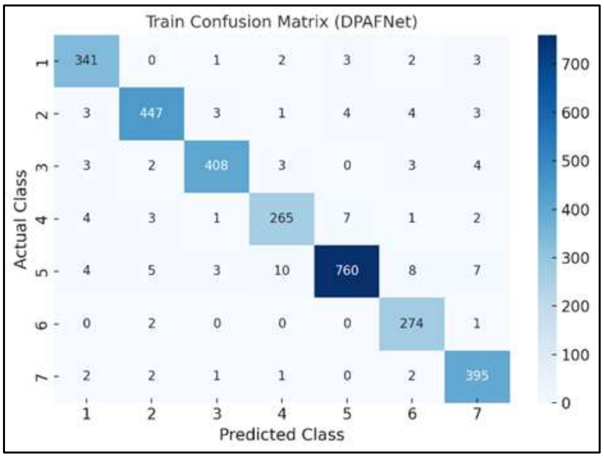
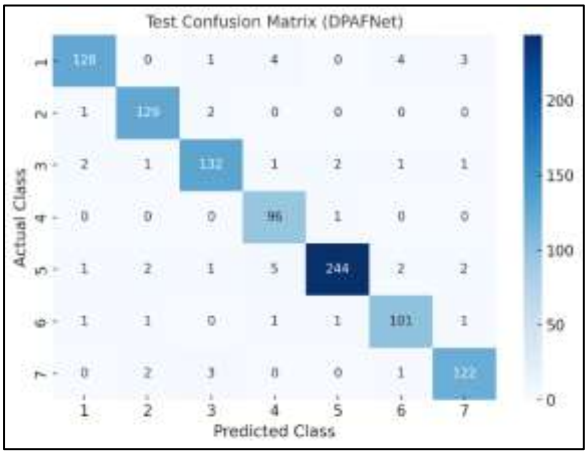


Figure 12. ROC Analysis of Proposed Model Testing

The confusion matrices elaborated in Figure 13 provide information about the effectiveness of the structural classification of the proposed DPAFNet model along the seven sections of diseases and pests in both the training and testing cases. During the training process shown in Figure 13a, the model classified most of the samples correctly with minimal cross-iteration. For example, Pseudostem Weevil reached the highest recognition rate, with 760 precise predictions made out of the total, followed by Black Sigatoka and Bacterial Soft Rot, each with over 400 precise predictions made, respectively.



(a) Training



(b) Testing

Figure 13. Confusion Matrix of Proposed Model **(a)** Training **(b)** Testing

Aphids and Yellow Sigatoka also performed well, with 341 and 395 true positives, respectively. Nonetheless, moderate misclassifications were observed in Panama Disease, with several segments being mislabeled as other disease categories due to exhibiting similar symptom variations and reduced class volume. Such performance integrity is demonstrated by DPAFNet during the testing stage, as shown in Figure 13b, where there is consistently high recognition performance for Scarring Beetle (244), Black Sigatoka (132), and Aphids (128). Although classifying Panama Disease remained more difficult with slight misclassification, the overall diagonal dominance of the two matrices indicates that the model performs at a satisfactory level. The results demonstrate that DPAFNet is reliable and can detect complex banana diseases.

As Table 3 shows, the overall outcome of the training performance of the proposed DPAFNet model involved the outstanding capacity of the model to classify banana leaf diseases across various classes. The model had an overall accuracy of 98.37%, a precision of 0.9544, recall of 0.8541, an F1-score of 0.8892, and an MCC of 0.8939, demonstrating good learning capability and balanced results in the predictions. Class-specific metrics illuminate perfect classification of Scarring Beetle and Aphids, with mean F1-scores of 0.9923 and 0.9964, respectively. The outcomes show that the rates of false positives and false negatives are low, particularly in those categories that are well represented and visually differentiated. The metrics of classes like Black Sigatoka and Yellow Sigatoka were also maintained stably, which confirms that precision and generalization at the class level are robust. Panama Disease, however, had a significantly low recall (0.255) and F1-score (0.4065), indicating challenges in detecting this class because it was not sampled as much as others and due to its visual similarities with other infections. Table 4 presents a summary of the testing performance, which proves the utility of the DPAFNet to generalize well to unseen data, with a maximum overall accuracy of 98.18%, precision of 0.9496, recall of 0.8386, and F1-score of 0.8844. Black Sigatoka (0.9066) and Pseudostem Weevil (0.9493) displayed noteworthy testing accuracy and F1-scores, although Panama Disease, once again, exhibited low recall (0.188), indicative of the model tending to be more conservative toward uncertain cases.

Table 3. Training Performance of Proposed DPAFNet Model

Class	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**MCC**
APHIDS	0.9989	0.993	1.000	0.9964	0.9967
BACTERIAL SOFT ROT	0.9621	0.9195	0.884	0.9006	0.8761
BLACK SIGATOKA	0.9855	0.9264	0.9022	0.9132	0.9164
PANAMA DISEASE	0.9879	1.000	0.255	0.4065	0.5021
PSEUDOSTEM WEEVIL	0.9482	0.9413	0.9812	0.9609	0.9212
SCARRING BEETLE	0.9991	0.9946	0.9901	0.9923	0.9918
YELLOW SIGATOKA	0.9893	0.9073	0.9012	0.9043	0.8922
Overall	0.9837	0.9544	0.8541	0.8892	0.8939

Table 4. Testing Performance of Proposed DPAFNet Model

Class	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**MCC**
APHIDS	0.9993	1.000	0.9931	0.9965	0.9961

BACTERIAL SOFT ROT	0.9647	0.9361	0.8665	0.8999	0.8699
BLACK SIGATOKA	0.9869	0.8735	0.9428	0.9066	0.8852
PANAMA DISEASE	0.9814	1.000	0.188	0.3164	0.4363
PSEUDOSTEM WEEVIL	0.9576	0.9278	0.9721	0.9493	0.8932
SCARRING BEETLE	0.9993	1.000	0.9847	0.9923	0.9895
YELLOW SIGATOKA	0.9912	0.9087	0.8421	0.8742	0.8647
Overall	0.9818	0.9496	0.8386	0.8844	0.8786

Further to validate the proposed model performance, conventional deep learning algorithms like ResNet50, VGG16, EfficientNetB0, DenseNet121, and MobileNetV2 are considered for comparative analysis. The simulation hyperparameters used for the existing models experimentation are presented in Table 5.

Table 5. Simulation Hyperparameters of Existing Models

S.No	Algorithm	Parameter	Type/Range
1	ResNet50	Learning Rate	0.001
2		Batch Size	64
3		Epochs	60
4		Optimizer	Adam
5	VGG16	Learning Rate	0.0005
6		Batch Size	32
7		Epochs	60
8		Optimizer	SGD with Momentum (0.9)
9	EfficientNetB0	Learning Rate	0.0007
10		Batch Size	32
11		Epochs	60
12		Optimizer	RMSProp
13	DenseNet121	Learning Rate	0.0001
14		Batch Size	64
15		Epochs	60
16		Optimizer	Adam
17	MobileNetV2	Learning Rate	0.0005
18		Batch Size	32
19		Epochs	60
20		Optimizer	Adam
21	ConVNeXtTiny	Learning Rate	0.0003
22		Batch Size	32
23		Epochs	100
24		Optimizer	Spotted Hyena Optimizer
25		Dropout Rate	0.3

26		Patch Size (Swin Transformer)	4
27		Window Size	7

Figure 14 illustrates the comparison graph in precision between the proposed DPAFNet model and existing deep learning models, demonstrating the strong performance of the proposed DPAFNet model over other models across 100 training epochs. As early as the first stages, DPAFNet shows a drastic and stable convergence to a precision that approaches 0.949, which is higher than all benchmark models over the learning curve. Interestingly, both ConvNeXtTiny and EfficientNetB0 showed competitive results regarding their precision, achieving around 0.91 and 0.89, respectively; however, they lagged behind the other models in terms of their overall course and post-optimization performance. DenseNet121 and ResNet50 were close contestants, while VGG16 was observed to be substantially lower, peaking at the 0.85 mark or below, which indicates structured simplicity and a lack of availability to dig deeper into feature hierarchies.

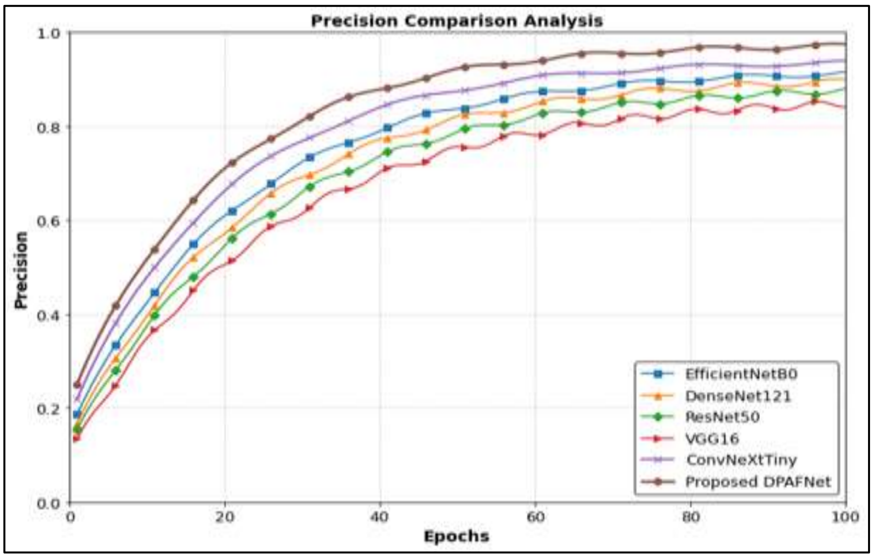


Figure 14. Precision Comparative Analysis

Figure 15 shows the recall comparison of the proposed DPAFNet model with existing DL models. Throughout the total training epochs of 100, DPAFNet has consistently dominated the top position in terms of recall performance, finally reaching its peak performance of about 0.911, indicating it is very adept at identifying correct cases and limiting incorrect ones. Other "closely ranked" models like ConvNeXtTiny and EfficientNetB0 follow, with all the models converging to nearly the same recall values of around 0.89 and 0.86, respectively. Meanwhile, DenseNet121 performs moderately at about 0.83, while ResNet50 and VGG16 are lower, with recall values of about 0.81 and 0.78, respectively. The remarkable strength of DPAFNet is associated with its dual-stream structure, which is based on MaxViT to learn globally receptive fields and HorNet-S to learn details locally.

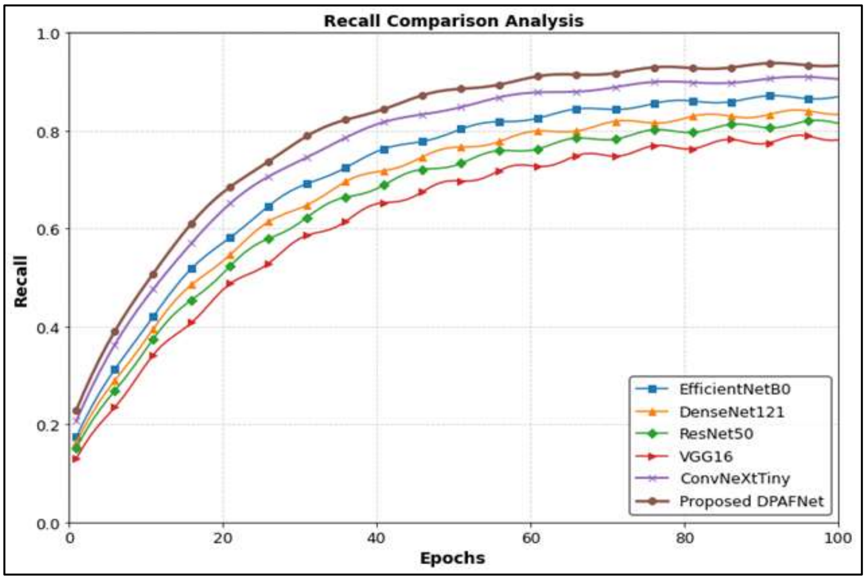


Figure 15. Recall Comparative Analysis

As Figure 16 shows, the evolution of the F1-Score gives a clear picture of how well the suggested DPAFNet mechanism balances precision and initiative during learning. DPAFNet is outperforming its counterparts as the modal position across all models and translates to the highest F1-Score of about 0.930, representing the ability of DPAFNet to sustain a low false-positive and false-negative rate. ConvNeXtTiny and EfficientNetB0 come next and plateau at 0.89 and 0.87, respectively, proving to have reasonable but somewhat lower classification uniformity. DenseNet121 and ResNet50 converge to lower values of approximately 0.84 and 0.81, while VGG16 has the lowest performance of slightly less than 0.79, indicating the model's failure to adapt to complicated visual information.

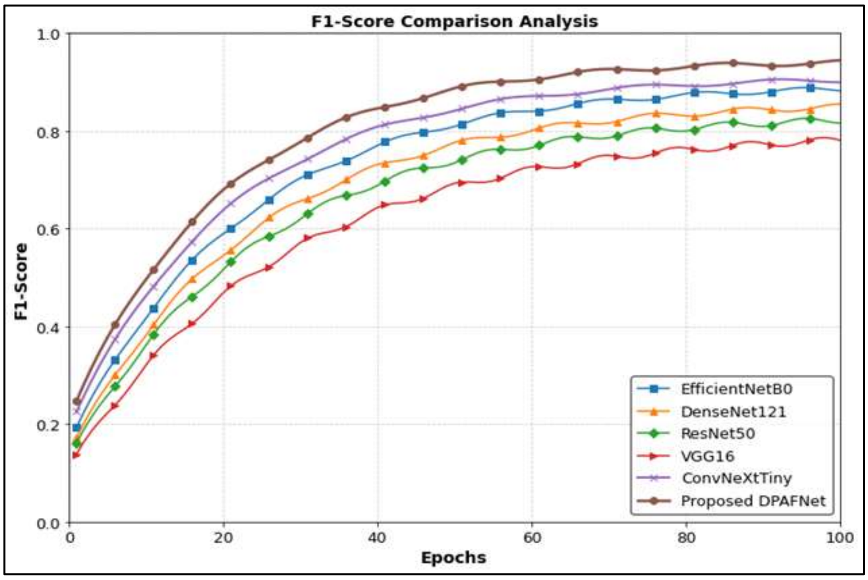


Figure 16. F1-Score Comparative Analysis

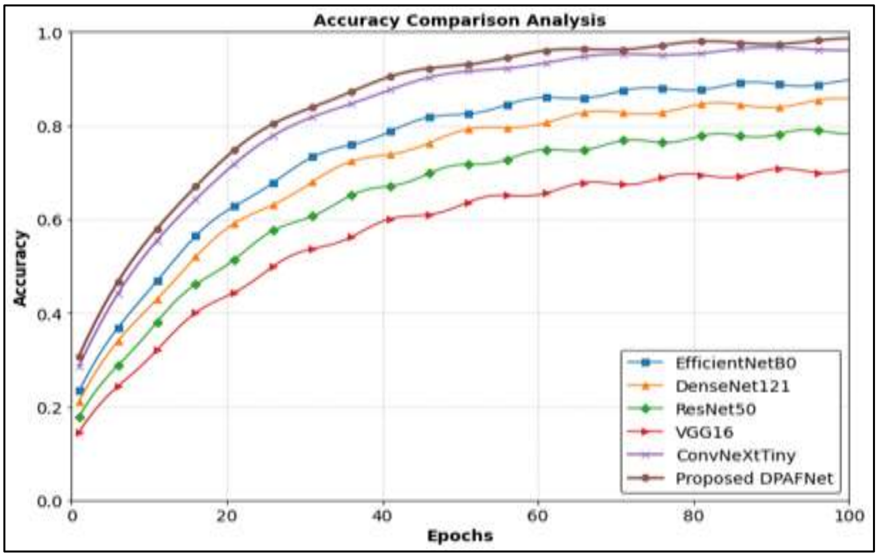


Figure 17. Accuracy Comparative Analysis

Figure 17 shows the relative accuracy improvement of the proposed DPAFNet model compared with other state-of-the-art deep learning architectures. The DPAFNet model stands at the top of the accuracy rankings with the highest performance of about 0.986. This outcome validates the strong performance of the proposed model in predicting disease categories as accurately as possible. The accuracies of ConvNeXtTiny and EfficientNetB0 are 0.977 and 0.950, respectively, whereas DenseNet121 and ResNet50 stabilize at lower levels, close to 0.935 and 0.920, respectively. VGG16 has the worst learning curve at 0.900, which is lower than that of the proposed model. Achieving high levels of accuracy with DPAFNet is based on its hybrid architecture, with MaxViT providing wide spatial coverage for feature extraction, and HorNeT-S facilitating the extraction of localized texture information that allows for precise differentiation between similarly appearing disease types. Such a dual-path fusion achieves scale-rich feature representation. In addition, the combination of the Adaptive Quantum Monarch Butterfly Optimization (AQMBO) allows for adapting the learning rate and optimizing parameters for convergence and parameter tuning more effectively.

Table 6. Performance Comparative Analysis of Proposed and Existing Models

S.No	Model	Precision	Recall	F1-Score	Accuracy
1	EfficientNetB0	0.913	0.865	0.875	0.950
2	DenseNet121	0.896	0.832	0.847	0.935
3	ResNet50	0.874	0.808	0.825	0.920
4	VGG16	0.857	0.778	0.795	0.900
5	ConvNeXtTiny	0.940	0.891	0.910	0.977
6	Proposed DPAFNet	0.949	0.911	0.930	0.986

A comprehensive comparison of the proposed DPAFNet model with existing deep learning models is given in Table 6 in terms of various measures. As a general trend, DPAFNet performs better than other existing models, achieving the best scores of precision (level 0.949 vs. 0.800), recall (0.911 vs. 0.715), F1-score (0.930 vs. 0.750), and accuracy (0.986 vs. 0.827).

These findings support the potential of the model to achieve a balance between sensitivity and specificity, with the lowest rates of false negatives and false positives. ConvNeXtTiny achieves an F1-score of 0.910 and accuracy of 0.977, which proves its architectural power, although it has not yet attained the more detailed learning capacity of DPAFNet. Other models, such as EfficientNetB0 and DenseNet121, are average, recording accuracies of 0.950 and 0.935, respectively. In contrast, ResNet50 and VGG16 fall short in terms of recall and low F1-scores. This highlights the shortcomings of these models in the analysis of complex and multi-class variations of diseases. DPAFNet has the advantage of a dual-path structure, combining the global vision attention of MaxViT with the hierarchical convolutional features of HorNet-S. Additionally, the Adaptive Quantum Monarch Butterfly Optimization (AQMBO) provides a refined autotuned selection of hyperparameters that optimizes the model training process. This design enables DPAFNet to present a high-performance and extensible system for banana disease recognition in practice.

The AQMBO-enhanced DPAFNet training cost is dominated by the time required to evaluate a population of P hyperparameter candidates over T AQMBO iterations; each candidate requires a validation run of E epochs, providing an effective search cost of $O(T \cdot P \cdot E)$ model evaluations as compared to $O(E)$ in a single conventional run. The AQMBO meta-update per iteration (migration plus quantum perturbation over d tuned variables) incurs $O(P \cdot d)$ in terms of arithmetic, which is negligible with respect to model evaluation (see the AQMBO formulation and hyperparameter vector in Eqs. (39) to (46)). Empirically, the tuned configuration has better validation accuracy early in training, surpassing ~94.5% in the first ~20 epochs and leveling off at 98.3% around epoch 100 (Fig. 7), suggesting faster convergence that compensates for the overheads of the search; this is confirmed by the monotonic drop in training/validation loss (Fig. 8).

5. Conclusion

A novel deep learning architecture DPAFNet was proposed in this research for robust banana leaf disease identification. The proposed model incorporates a learnable preprocessing module, a dual-path feature extractor utilizing MaxViT and HorNet-S, and a cross-layer attention fusion (CLAF) mechanism. Additionally, optimization is done using the Adaptive Quantum Monarch Butterfly Optimization (AQMBO) algorithm. The proposed model was trained and evaluated using a banana disease dataset from Mendeley comprising seven disease categories and one healthy class. Extensive experiments were conducted and the proposed DPAFNet achieved a test accuracy of 0.986, precision of 0.949, recall of 0.911, and F1-score of 0.930 which is better than models such as EfficientNetB0 (accuracy: 0.950), DenseNet121 (0.935), and ConvNeXtTiny (0.977). Precision-recall and ROC analysis further demonstrated the model’s robustness in recognizing underrepresented classes like Panama Disease. Despite these advancements, the model's recall on minority classes remains a challenge, indicating sensitivity to class imbalance. Future work could explore advanced data augmentation, synthetic data generation, and lightweight transformer integration for real-time deployment in mobile agricultural advisory systems. Overall, DPAFNet represents a scalable and accurate solution for smart agriculture and early disease intervention.

References

[1] Ismaila, Abubakar Abubakar, Khairulmazmi Ahmad, Yasmeen Siddique, Muhammad Aswad Abdul Wahab, Abdulaziz Bashir Kutawa, Adamu Abdullahi, Syazwan Afif Mohd Zobir, Arifin Abdu, and Siti Nor Akmar Abdullah. "Fusarium wilt of banana: current update and sustainable disease control using classical and essential oils approaches." Horticultural Plant Journal 9, no. 1 (2023): 1-28.

[2] Bischoff, Vinicius, Kleinner Farias, Juliano Paulo Menzen, and Gustavo Pessin. "Technological support for detection and prediction of plant diseases: A systematic mapping study." Computers and Electronics in Agriculture 181 (2021): 105922.

[3] Sk Mahmudul Hassan, Khwairakpam Amitab, Michal Jasinski, Zbigniew Leonowicz, Elzbieta Jasinska, Tomas Novak, Arnab Kumar Maji, “A Survey on Different Plant Diseases Detection Using Machine Learning Techniques,” Electronics, vol. 11, no. 17, (2022):1-29.

[4] Nixon Jiménez, Stefany Orellana, Bertha Mazon-Olivo, Wilmer Rivas-Asanza, Iván Ramírez-Morales, “Detection of Leaf Diseases in Banana Crops Using Deep Learning Techniques,” AI, vol.6, no.3, (2025):1-27.

[5] Samuel Manoharan, J., Braveen, M. & Ganesan Subramanian, G, “A hybrid approach to accelerate the classification accuracy of cervical cancer data with class imbalance problems,” International Journal of data mining, vol. 25, no.3/4, (2022):234 – 259.

[6] Kevin Yan, Md Kamran Chowdhury Shisher, Yin Sun, “A Transfer Learning-Based Deep Convolutional Neural Network for Detection of Fusarium Wilt in Banana Crops,” AgriEngineering, vol. 5, no. 4, (2023): 2381-2394.

[7] Manojkumar Patel, “Deep Learning based Automated System for Banana Plant Disease Detection and Classification. International Journal of Next-Generation Computing, vol.15, no.2, (2024):1-16.

[8] Ilham Rahmana Syihad, Muhammad Rizal, Zamah Sari, Yufis Azhar, “CNN Method to Identify the Banana Plant Diseases based on Banana Leaf Images by Giving Models of ResNet50 and VGG-19,” Journal RESTI, vol.7, no. 6, (2023):1309-1318.

[9] Huanshuo Zhang, Guobiao Ren, “Intelligent leaf disease diagnosis: image algorithms using Swin Transformer and federated learning,” The Visual Computer, vol. 41, no. 7, (2024): 4815 - 4838.

[10] Ebru Ergün, “High precision banana variety identification using vision transformer-based feature extraction and support vector machine,” Scientific Reports, vol.15, pp. (2025):1-16.

[11] Nidhi Malik, Rita Chhikara, Gaurangna Yadav, Preet Sharma, Navdeep Sisodia, “Advanced Deep Learning techniques for Cauliflower Plant Disease Detection,” Procedia Computer Science, vol. 259, (2025):1326-1335.

[12] Jinsheng Deng, Weiqi Huang, Guoxiong Zhou, Yahui Hu, Liujun Li, Yanfeng Wang, “Identification of banana leaf disease based on KVA and GR-ARNet,” Journal of Integrative Agriculture, vol. 23, no. 10, (2024): 3554-3575.

[13] Christian A. Elinisa, Neema Mduma, “Mobile-Based convolutional neural network model for the early identification of banana diseases,” Smart Agricultural Technology, vol.7, (2024):1-10.

[14] Md. Abdullahil Baki Bhuiyan, Hasan Muhammad Abdullah, Shifat E. Arman, Sayed Saminur Rahman, Kaies Al Mahmud, “BananaSqueezeNet: A very fast, lightweight convolutional neural network for the diagnosis of three prominent banana leaf diseases,” Smart Agricultural Technology, vol. 4, (2023):1-13.

[15] Tejaswini, Priyanka Rastogi, Swayam Dua, Manikanta, Vikas Dagar, “Early Disease Detection in Plants using CNN,” Procedia Computer Science, vol. 235, (2024):3468-3478.

[16] Xinyu Dong, Qi Wang, Qianding Huang, Qinglong Ge, Kejun Zhao, Xingcai Wu, Xue Wu, Liang Lei, Gefei Hao, “PDDD-PreTrain: A Series of Commonly Used Pre-Trained Models Support Image-Based Plant Disease Diagnosis,” Plant Phenomics, vol. 5, (2023):1-20.

[17] Mahadevan.K, A. Punitha, J. Suresh, “A novel rice plant leaf diseases detection using deep spectral generative adversarial neural network,” International Journal of Cognitive Computing in Engineering, vol. 5, (2024): 237-249.

[18] Xiaojun Xie, Fei Xia, Yufeng Wu, Shouyang Liu, Ke Yan, Huanliang Xu, Zhiwei Ji, “A Novel Feature Selection Strategy Based on Salp Swarm Algorithm for Plant Disease Detection,” Plant Phenomics, vol.5, (2023): 1-17.

[19] Pudumalar.S, Muthuramalingam.S, “Hydra: An ensemble deep learning recognition model for plant diseases,” Journal of Engineering Research, vol. 12, no. 4, (2024): 781-792.

[20] Aishwarya MP, Padmanabha Reddy, “Ensemble of CNN models for classification of groundnut plant leaf disease detection,” Smart Agricultural Technology, vol. 6, (2023):1-14.

[21] Sunil S. Harakannanavar, Jayashri M. Rudagi, Veena I Puranikmath, Ayesha Siddiqua, R Pramodhini, “Plant leaf disease detection using computer vision and machine learning algorithms,” Global Transitions Proceedings, vol. 3, no 1, (2022): 305-310.

[22] Vinay K, Vempalli Surya, Thushar S, Tripty Singh, Apurvanand Sahay, “A Deep Learning Framework for Early Detection and Diagnosis of Plant Diseases,” Procedia Computer Science, vol. 258, (2025): 1435-1445.

[23] Kadambari Raghuram, Malaya Dutta Borah, “A Hybrid Learning Model for Tomato Plant Disease Detection using Deep Reinforcement Learning with Transfer Learning,” Procedia Computer Science, vol. 252, (2025): 341-354.

[24] Jiuqing Dong, Yifan Yao, Alvaro Fuentes, Yongchae Jeong, Sook Yoon, Dong Sun Park, “Visual information guided multi-modal model for plant disease anomaly detection,” Smart Agricultural Technology, vol. 9, (2024):1-13.

[25] https://data.mendeley.com/datasets/4wyymrcpyz/1