SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Explainable Hybrid CNN–Vision Transformer Framework For Robust Moringa Leaf Disease Diagnosis Using U-Net Segmentation And Yolov8 Localization

Sivamurugan Chelladurai[1*], Perumal Balasubramani[2], Santhi Kumaran[2]

[1*] Research Scholar, Department of Electronics and Communication Engineering, Kalasalingam Academy of Research and Education, Virudhunagar, Tamilnadu, India, Email: sivaapril1704@gmail.com
[2] Associate Professor, Department of Electronics and Communication Engineering, Kalasalingam Academy of Research and Education, Virudhunagar, Tamilnadu, India, Email:perumal@klu.ac.in
[2] Professor, Department of Computer Engineering, The Copperbelt University, Kitwe, Zambia, Email: santhi.kr@cbu.ac.zm

Abstract— Moringa oleifera is a plant species that is economically, nutritively, and medicinally relevant, where its productivity is adversely impacted by leaf infections caused by bacteria and fungi. Early detection of such disease infection in plants is crucial for ensuring better yield and supporting precision agriculture; but the CNN models used for detecting these diseases have limited ability in capturing global context relationships and are not explainable in addition to showing poor performance in a variety of environmental settings. In this study, we propose a hybrid deep learning model for moringa leaves classification that makes use of semantic segmentation using U-Net, ResNet50, and Vision Transformer (ViT). The proposed model involves the process of image processing and semantic segmentation of disease-infected regions, followed by hybrid feature extraction that takes advantage of discriminative local spatial features extracted from the image by ResNet50 and long-range context features extracted by ViT. Moreover, YOLOv8 is used for detecting the diseased parts in the moringa leaves and the Grad-CAM based Explainable Artificial Intelligence (XAI) technique provides the visualization support to make the model more transparent and reliable. Experimental results show that the proposed model achieves 99.68% classification accuracy, 99.54% precision, 99.47% recall, 99.50% F1-score, 99.81% specificity and an AUC of 0.998 which are superior to ResNet50, VGG16, DenseNet121 and Vision Transformer alone. The visualization results of Grad-CAM have shown that the model is able to focus on the biologically meaningful regions in the moringa leaves affected by diseases.

Keyword: Plants, Agriculture, Result, Model, Disease, Model.

1. Introduction

Agriculture is very significant for securing the global food supply, economic growth, and sustainable livelihood, especially in the case of developing nations, where the production of crops becomes the mainstay of the local economy. Out of all economically important crops, one such crop that has become quite popular due to its outstanding nutritional, medicinal, and industrial properties is Moringa oleifera. It has been found to be abundant in various nutrients such as vitamins, minerals, antioxidants, protein, and other bioactive substances, making moringa quite a resourceful crop for use in the food, pharmaceutical, cosmetic, and nutraceutical industries [1]. But the productivity and quality of moringa crops are quite influenced by bacterial and fungal leaf diseases that affect the photosynthesis process, growth of the plant, and ultimately reduce the output of the crops [2].

Conventionally, the process of diagnosing diseases in plants involves manual visual examination done by farmers or agricultural professionals. Even though this method is highly prevalent, the accuracy of this process largely relies on the competence of individuals involved, hence making it a highly subjective, resource-intensive, and time-consuming process especially in large-scale farming operations [3]. In the recent past, there have been advancements through the use of artificial intelligence (AI), computer vision, and deep learning, which have led to huge success in the field of automated plant disease identification. CNN has become an important technology since it allows automatic extraction of image features hierarchy-wise without needing feature engineering. VGG16, ResNet, DenseNet, and Inception are always the leading

architectures when it comes to classification of accuracy in the various data sets used for plant disease classification [4], [5].

Despite the progress made, there are still some issues that need to be resolved. Traditional CNN designs focus on learning the local spatial structures and do not consider long-range contextual relationships spread all over the leaf image. Therefore, diseases that have similar color distribution, complex lesions, or infections scattered all over the leaf could not be distinguished properly. Although Vision Transformer designs have shown great promise for modelling the global context information by applying self-attention techniques, the problem is that they usually require huge annotated data sets and computational power, which makes it impossible to deploy in agriculture applications [6]. Moreover, many studies classify disease directly from the original images without performing segmentation beforehand, causing the background pixels to interfere in the process and decrease the accuracy of classification. Another drawback is that current studies usually classify or localize the disease separately, whereas very few incorporate localization, classification, and explainability into one system. Lastly, low interpretability is another issue that limits the adoption of deep learning models because they are hard to interpret [7], [8].

Despite significant advancements in the field of deep learning based plant disease diagnosis, a number of key research questions still require investigation. The literature review revealed that previous research on the topic of plant disease diagnosis, especially of Moringa oleifera, mostly examines semantic segmentation, disease diagnosis, localization, and explainable artificial intelligence individually. Therefore, it appears that there is still room for the development of comprehensive frameworks able to provide accurate, robust, and explainable disease diagnosis in practice. Another issue that needs further research is the combination of the ability of local feature learning provided by CNNs and the ability of global context modeling performed by transformers for the problem of moringa leaf disease diagnosis.

In contrast to the previously suggested frameworks that only integrate CNN and Transformer architectures, the proposed framework incorporates semantic segmentation, hybrid local and global feature learning, disease localization, and Explainable Artificial Intelligence (XAI). First of all, the proposed framework performs segmentation of the infected regions in the original image in order to decrease the impact of background information on further processing. ResNet50 is used to extract the local features of the image, whereas Vision Transformer takes advantage of multi-head self-attention mechanism in order to learn the dependencies in the image context. Moreover, YOLOv8 is used to detect disease-affected regions, and Grad-CAM-based XAI is applied to provide the explanation of the model decisions visually. In order to increase the robustness and generalization abilities of the proposed framework, training is performed on the images collected from the field and augmented data samples. In contrast to previous works, where the problems of classification, localization, or explainability of disease were addressed separately, the proposed framework integrates all these elements into a single end-to-end system.

The main contributions made by this research work are stated below:

- An Explainable end-to-end system based on the combination of U-Net semantic segmentation, ResNet50, Vision Transformer, YOLOv8, and Grad-CAM is designed to detect diseases accurately in moringa leaves.
- The U-Net based semantic segmentation system is included to localize the infected parts of leaves before classification to make the disease identification more efficient.
- The YOLOv8 model is used for localizing the disease affected regions in the leaves to make the process of classification and localization both at once.
- The Explainable Artificial Intelligence through Grad-CAM is utilized to make the predictions transparent.
- Multi-Dataset Learning methodology based on the collection of images from real-world farms and augmentations is implemented in this research work.

As a whole, the suggested approach creates a precise, resilient, and interpretable end-to-end intelligent system for diagnosing diseases that can be used for precision agriculture and smart farming in the next generation. The rest of the paper is structured in the following way. Section 2 provides the literature review of the plant disease detection problem. Section 3 gives a review of the current deep learning approaches examined in this paper. Section 4 describes the proposed hybrid approach. Section 5 shows the obtained results and their comparison. Section 6 provides a discussion of the results. Finally, Section 7 concludes the paper.

2. Related work

2.1 Traditional Machine Learning and Early Deep Learning Approaches

Automated detection of plant disease has witnessed substantial advancement from conventional image processing approach to more recent deep learning techniques. The early approaches were based mostly on

the use of manually designed features such as color, texture, and shape features, along with machine learning classifiers like SVM, ANN, RF, DT, and k-NN. While these approaches lowered the dependency on visual inspections, they still exhibited high reliance on the design of features and image acquisition conditions. Studies conducted by Pydipati et al. [9], Al-Hiary et al. [10], Arivazhagan et al. [11], Phadikar and Sil [12], Camargo and Smith [13], and Revathi and Hemalatha [14] have shown that color, texture, and statistical features can be successfully employed in detecting plant disease under controlled laboratory conditions. However, these approaches showed high sensitivity to illumination conditions, background environment, leaf orientations, and severity of disease.

Deep learning revolutionized plant disease identification due to its capability to automatically extract hierarchical features directly from images. Deep CNNs are different from other machine learning methods because they do not need manual feature engineering; rather, CNNs learn discriminative features through end-to-end optimization. For the first time, Mohanty et al. [15] showed that the deep CNN architectures of AlexNet and GoogLeNet achieved high accuracy in disease classification compared to traditional machine learning approaches, using the PlantVillage dataset. Then Ferentinos [16] used deep CNN architectures for recognizing multi-crop diseases, and he found that CNNs achieve high classification accuracy in multiple crop species. Similarly, Too et al. [17] performed comparison studies on pre-trained CNN architectures and showed that transfer learning greatly helped in improving the classification accuracy of CNNs and also reduced their training time along with the need for large agriculture data sets. However, earlier CNN architectures did not focus much on localization, segmentation, and explainability of plant disease images.

A number of reviews have also pointed out the strengths and challenges of deep learning in agriculture. According to Kamilaris and Prenafeta-Boldú [18], CNN-based models were considered the dominant methods for agricultural image processing owing to their excellent ability to learn features as well as scalability. Similarly, Barbedo [19] conducted a critical review of the applications of deep learning in plant disease diagnosis and indicated a number of challenges associated with data set limitations, poor field validation, explainability, and low generalizability within agricultural settings. Overall, these studies confirm that despite the significant improvements in disease detection accuracy due to the application of deep learning, the earlier CNN models were mainly concerned with image classification rather than disease localization and segmentation.

While feature-based methods showed promising results in disease classification in a laboratory setting, the need for manually engineered features and their susceptibility to changes in environmental conditions were among their shortcomings that precluded their scalability and robustness. In turn, these shortfalls prompted the research community to shift towards deep learning methods which have the ability to learn their own discriminative feature representations. Overall, these studies made CNNs the prevailing approach for automated plant disease diagnosis due to their powerful feature learning and classification capabilities. Nevertheless, initial CNN-based works mostly worked on controlled datasets and provided limited functionality related to lesion localization, semantic segmentation, model interpretability, and robustness in practice. Despite the fact that deep learning methods significantly enhanced the feature representation and classification capabilities, some issues such as global context learning, explainability, lesion localization, and robustness in practice remained open.

2.2 CNN-Based Plant Disease Classification

The effectiveness of CNNs has greatly improved automatic plant disease detection through end-to-end feature learning, eliminating the need for handcrafted feature extraction. In comparison with traditional ML algorithms, CNN is capable of automatically learning features from low-level edges/texture features up to higher-level semantic disease features, thus enhancing the classification performance and robustness. This has led to many different types of CNN being explored for recognizing plant diseases, such as VGG16, ResNet, DenseNet, MobileNet, EfficientNet, and attention mechanisms-based CNN. Of all these CNNs, one of the most popular transfer learning architectures in use is VGG16 introduced by Simonyan and Zisserman [20]. Its simplicity and effectiveness as an architectural structure using stacked 3×3 convolutional layers has enabled its successful application in many different datasets of crop diseases. The downside to this network is that the large number of trainable parameters makes it computationally costly and memory-consuming, thus making its real-time deployment problematic. To mitigate the optimization issues caused by deep architectures, He et al. [21] came up with the Residual Network (ResNet) in which identity shortcut connections make it possible to address the vanishing gradient problem for effectively training much deeper architectures.

Additional advancements have been made through the adoption of DenseNet, presented by Huang et al. [22]. In DenseNet, dense connectivity between layers is established, where dense connection facilitates feature reuse and gradient propagation. As a result, DenseNet is highly efficient in reducing redundancy in feature learning and improves performance in classification tasks, especially in identifying complicated disease patterns. However, the dense connectivity architecture requires high memory consumption during

training that is a drawback since DenseNet cannot be applied in resource-limited agricultural devices. In order to minimize computational complexity, there have been attempts in developing lightweight models including MobileNet [23] and EfficientNet [24]. MobileNet makes use of depthwise separable convolution in order to minimize computational cost, while EfficientNet makes use of compound scaling of network depth, width, and resolution. The combination of those two enables the creation of efficient models while still maintaining decent classification accuracy for mobile and edge-based agricultural applications. Although transformer-based models are advancing rapidly nowadays, CNNs are commonly used in agricultural image analysis due to their inductive biases and efficiency and great performance on relatively small labeled datasets for plant disease detection.

The latest studies have considered attention mechanism and feature refinement approaches to improve the CNN results. Hu et al. [25] developed Squeeze-and-Excitation (SE) Networks, where the network learns to adjust the feature responses of channels for better discriminatory capabilities. The Convolutional Block Attention Module (CBAM) was proposed by Woo et al. [26], where the module considers the channel attention and spatial attention to highlight the regions related to disease images and ignoring the rest of the unnecessary data. Through the attention mechanisms, CNN can focus on disease-related regions and ignore background data.

Some comparative research has also been carried out to analyze the efficiency of the use of pretrained CNN models in agriculture. According to Too et al. [27], ResNet50 and DenseNet121 always outperformed conventional CNN models due to their higher ability of feature propagation and transfer learning capacity. Similarly, according to Ferentinos [28], the classification accuracy using deep CNN models was significantly high among various crop species; but their performance decreased when the tests were performed under practical agricultural settings due to the differences in lighting, occlusion, and the severity of diseases. Also, according to Barbedo [29], even though CNNs greatly help in achieving automated disease detection, the majority of the existing literature only focuses on the image classification without involving lesion localization, semantic segmentation, or explainable artificial intelligence in a single framework. Thus, through these comparative studies, it is clear that CNN architectures perform very well in terms of classification accuracy in benchmark settings. However, the generalization capacity in real agricultural settings is limited by illumination, occlusion, severity of diseases, and background complexity.

Even as CNN-based models have shown remarkable performance results, they have their own shortcomings. While CNNs perform excellently in extracting hierarchical local features, their receptive fields are limiting the modelling of the long-range contextual dependencies between distributed disease symptoms. Thus, diseases with irregular lesion distribution and highly similar visual characteristics cannot be distinguished using convolution operations only. In addition, most CNN-based models have been tested on benchmark datasets collected under real-world agricultural environments but little is known about how they perform in real-world agricultural environments. This is one of the reasons why transformer-based models that can model global contextual information through self-attention were developed as mentioned below.

2.3 Vision Transformer and Hybrid CNN–Transformer Models

While Convolutional Neural Networks (CNN) have shown impressive results in the task of classifying plant diseases, the convolution operations used by CNNs extract only the localized features, making it difficult for them to understand the relationships between symptoms spread over the whole leaf. Transformer models have been introduced to overcome such limitations using self-attention operations to create global interactions between image patches. While CNNs only learn local context, Vision Transformers learn both the local and global contexts together.

The groundbreaking study of Dosovitskiy et al. [30] resulted in the creation of the Vision Transformer (ViT). In this approach, the input image is divided into fixed-sized patches and transformed using transformer encoder blocks. This architecture showed competitive performance in image classification by capturing long-range dependencies. The ViT approach needed a huge annotated dataset and high computational power for better performance. Thus, the use of ViT in agriculture is limited to datasets with fewer annotations. To address the issue of computational efficiency, Liu et al. [31] introduced the Swin Transformer. Swin Transformer uses shifted-window self-attention to capture hierarchical features while keeping the computational complexity low. The Convolutional Vision Transformer (CvT) was introduced by Wu et al. [32] through the integration of convolutional layers in transformers. Likewise, Graham et al. [33] designed the LeViT architecture by incorporating convolutional operations with transformers to get transformer-level accuracy with high computational efficiency. Thus, the evolution of these transformer-based architectures reflects that there has been an advancement towards improved computational efficiency, spatial representation, and data efficiency while maintaining the capacity of modeling long-range contextual dependencies. Despite all these advancements, most transformer-based architectures need relatively larger annotated datasets and computational power as compared to conventional CNNs.

A wide range of research has recently focused on hybrid CNN–Transformer models that utilize the complementary abilities of the CNN for extracting features and transformer for modeling the global context of the data. In such systems, the role of CNN is in extracting fine details of lesions' properties including texture, color, and edges, while transformers analyze the interactions between the distant parts of the diseases. Hybrid learning techniques are known to be more robust, informative and discriminative than the architectures based on either CNN or Transformer. Additionally, hybrid segmentation models like TransUNet [34] and UNETR [35] have been developed that incorporate transformer encoders into encoder-decoder architecture to improve semantic segmentation performance while maintaining local and global features. Though initially designed for medical images, these models proved to have promising results for agriculture segmentation due to their ability to detect irregular boundaries and patterns of the diseases.

Data-efficient approaches to improve the efficiency of transformers have been another focus of recent research efforts. The Data-efficient Image Transformer (DeiT) [36] has shown that, through knowledge distillation and optimization of training methods, less data can be used for transformer learning. In addition, the application of transformer models in detectors, such as DETR [37] by Carion et al., enabled reformulation of object detection by applying transformers, thus facilitating end-to-end detection of objects through transformer models without requiring any handcrafted region proposals. Motivated by these research findings, recent works in agriculture have employed hybrid CNN-Transformer frameworks for plant diseases classification and crop monitoring, achieving high levels of classification accuracy, robustness, and effective feature representation under varying environmental conditions [38], [39].

Despite all these improvements, there are several limitations to using transformer-based methods. Vision Transformers demand more computation power, longer training, and much larger annotated datasets than classical CNNs do. Moreover, current research is oriented on solving the problem of image classification, semantic segmentation, localization, or explaining the decision made independently from each other. This is the motivation for building an architectural design of a deep learning algorithm that can incorporate all these aspects into one diagnostic system, as described in the methodology section below.

2.4 Semantic Segmentation, Object Detection, Explainable Artificial Intelligence, and Research Gap

The development of deep learning technology has also allowed for the application of semantic segmentation, object detection, and explainable artificial intelligence (XAI) to disease detection in plants. All these techniques provide means for identifying infected areas, localizing symptoms, and interpreting the results of prediction, which allows us to enhance the accuracy and practicality of using artificial intelligence in agriculture. It is important to combine all these techniques in order to meet the requirements of precision agriculture, where disease classification needs to be visually verified.

Semantic Segmentation has been observed to be an effective approach in segregating the regions of interest before classifying the disease. As opposed to conventional image classification techniques in which one class is labeled to the whole image, semantic segmentation technique entails labeling of each pixel with a particular class to discriminate the diseased parts from the normal part of the image. The authors in [40] presented U-Net, an encoder-decoder network that exploits the power of skip connections with multi-scale feature learning to perform accurate segmentation. Later on, UNet++ developed by Zhou et al. [41] enhanced the performance of segmentation through nested skip connections that decrease the semantic gap between features of the encoder and the decoder. Likewise, the authors in [42] utilized attention gates within U-Net to highlight disease-related pixels and eliminate irrelevant background pixels. Recently, hybrid approaches like TransUNet [34] and UNETR [35] have been developed which combine transformer encoders with convolutional networks to capture local as well as global information for achieving better results in segmentation of images having complex lesion boundaries and disease patterns. However, most of the semantic segmentation models have been assessed independently of disease classification and explainability.

Object detection takes plant disease detection one step further by automating the process of finding infected areas through the use of bounding boxes. The early two-stage detectors had good performance on the localization of objects but were computationally expensive, which made it impossible to apply them in real time. With the introduction of the You Only Look Once (YOLO) detector by Redmon et al. [43], the time taken for the process was reduced greatly, thanks to the formulation of object detection as a regression task in a single stage. Further development of YOLO to YOLOv3 [44], YOLOv5 [45], and YOLOv8 [46] resulted in increased accuracy, increased inference speed, and reduced computation through innovations such as feature pyramid networks, anchor optimization, and anchor-free detection, among others. These models have been used successfully for disease detection in agriculture applications. However, the majority of the research involving the use of YOLO models have been mainly centered on lesion detection without the inclusion of semantic segmentation, feature explainability, and hybrid classifiers, which limits their diagnostics capabilities.

Moreover, the proliferation of deep learning approaches has led to the emergence of the need for Explainable Artificial Intelligence (XAI) in agriculture. Though deep neural networks deliver high-quality predictive results, their internal representations are quite complicated for interpretation. To address the issue, visualization methods, including Grad-CAM [47], LIME [48], and SHAP [49], were employed extensively to make predictions interpretable. While Grad-CAM creates class activation maps showing disease-related regions of an image which contribute to the final prediction, LIME and SHAP provide local explanations for individual predictions of the classifier based on features. The above method ensures interpretability and thus makes one confident in the system, knowing that predictions are based on biologically meaningful properties of the disease, and not on any background information. Thus, explainability is mostly separated from the entire diagnostic process, preventing it from being a part of improving reliability of the models.

However, although considerable progress has been achieved on such components as CNN classifiers, transformer frameworks, semantic segmentation, object detection, and XAI, current research typically tends to explore these aspects individually instead of integrating all these elements into a cohesive system that can help diagnose lung infection diseases. CNNs have proved efficient in modeling local lesion properties but do not possess good contextual understanding, while transformers have demonstrated their superiority in contextual understanding at the expense of higher computational requirements. Moreover, segmentation and object detection have shown high accuracy in identifying infected areas but have rarely communicated with classifier models to boost feature learning; explainability approaches allow for visualization without improvement in localization and classification.

Inspired by the above research gaps, the current work designs a comprehensive deep learning system that combines U-Net-based semantic segmentation, ResNet50 for extracting local features, Vision Transformer for extracting global context, YOLOv8 for disease localization, and Grad-CAM for making models explainable. In addition to that, the framework uses multi-dataset learning and data augmentation to ensure robustness and generalizability in various agricultural settings. With the use of all the above technologies in one integrated system, the current methodology attempts to deliver an effective and efficient approach to intelligent moringa leaf disease detection.

Table 1 Comparison of existing deep learning methods for plant leaf disease classification

Ref	Author & Year	Model Used	Dataset	Accuracy	Key Contribution	Limitation
[9]	Pydipati et al., 2006	Color Features + SVM	Citrus Leaves	93.0%	Early automated disease diagnosis using handcrafted features	Manual feature extraction and poor generalization
[10]	Al-Hiary et al., 2011	Image Processing + SVM	Plant Leaf Images	94.3%	Automatic disease detection using segmentation	Sensitive to illumination changes
[11]	Arivazhagan et al., 2013	Texture Features + ANN	Plant Leaves	94.7%	Combined texture and color descriptors	Performance affected by background complexity
[12]	Phadikar and Sil, 2008	Pattern Recognition	Rice Leaves	92.4%	Statistical feature extraction for rice disease diagnosis	Crop-specific implementation
[13]	Camargo and Smith, 2009	SVM	Leaf Dataset	95.1%	Improved disease classification using texture analysis	Limited scalability
[14]	Revathi and Hemalatha, 2012	ANN	Cotton Leaves	94.9%	Automated cotton disease identification	Poor robustness under field conditions
[15]	Mohanty et al., 2016	AlexNet, GoogLeNet	PlantVillage	99.35%	First large-scale CNN application in plant disease detection	Reduced performance on real-world images

[16]	Ferentinos, 2018	Deep CNN	Multi-crop Dataset	99.53%	High-accuracy multi-crop disease classification	No localization or explainability
[17]	Too et al., 2019	VGG16, ResNet50, DenseNet121	PlantVillage	99.75%	Comparative evaluation of transfer learning models	Limited field validation
[20]	Simonyan & Zisserman, 2015	VGG16	ImageNet / Transfer Learning	92.7%	Strong hierarchical feature extraction	High computational complexity
[21]	He et al., 2016	ResNet50	ImageNet	96.43%	Residual learning for deeper networks	Limited global contextual learning
[22]	Huang et al., 2017	DenseNet121	ImageNet	74.98%	Dense feature reuse and improved gradient flow	High memory consumption
[24]	Tan & Le, 2019	EfficientNet	ImageNet	84.3%	Efficient model scaling	Less effective for highly complex disease patterns
[30]	Dosovitskiy et al., 2021	Vision Transformer	ImageNet-21K	88.55%	Global contextual feature learning	Requires large datasets and GPUs
[31]	Liu et al., 2021	Swin Transformer	ImageNet	87.3%	Hierarchical transformer representation	Computationally intensive
[32]	Wu et al., 2021	Convolutional Vision Transformer	ImageNet	87.7%	CNN and transformer integration	Complex architecture
[34]	Chen et al., 2021	TransUNet	Medical Images	Dice: 89.7%	Hybrid CNN–Transformer segmentation	High computational cost
[43]	Redmon et al., 2016	YOLO	PASCAL VOC	63.4 mAP	Real-time disease localization	Bounding-box localization only
[47]	Selvaraju et al., 2017	Grad-CAM	ImageNet	—	Explainable AI visualization	Does not improve prediction accuracy

3. Existing deep learning models

Prior to discussing the proposed hybrid approach, some state-of-the-art deep learning architectures are analyzed to serve as a baseline for moringa leaf disease classification task. ResNet50, VGG16, DenseNet121, and Vision Transformer (ViT) models are chosen due to their popularity and distinctive feature learning approaches that include residual learning, deep convolutional feature extraction, dense feature propagation, and global contextual modeling using transformers, respectively. All of them have shown outstanding results in various tasks of image classification and agricultural image processing; therefore, they can be regarded as suitable benchmark architectures. At the same time, all four architectures suffer from intrinsic disadvantages concerning local feature learning, computational requirements, or contextual representation that led to the need for developing the proposed hybrid approach. In addition, while ResNet50 serves as an independent benchmark architecture, its residual learning ability is utilized in the proposed architecture as a backbone for a convolutional part in combination with Vision Transformer.

3.1 ResNet50

ResNet50 [50] is a deep residual convolutional neural network architecture, introduced by He et al., that helps in solving the degradation problem that occurs with deeper networks. ResNet50 uses residual learning using residual blocks with identity shortcuts, which enables efficient propagation of feature

information and gradients over multiple layers. This architecture comprises an initial convolutional layer and four residual stages consisting of bottleneck residual blocks, resulting in fifty weighted layers altogether. Every bottleneck block involves 1×1, 3×3, and 1×1 convolutions and identity mapping to preserve low-level information while learning high-level features. The ResNet50 architecture helps to learn discriminative hierarchical features like lesion borders, texture differences, and colour changes, which are characteristics of plant diseases through the residual learning technique. Therefore, ResNet50 has emerged as one of the most widely used backbone architectures in plant disease classification, medical imaging, and object recognition tasks due to its good convergence properties, computational efficiency, and transfer learning abilities. ResNet50 is chosen as the baseline architecture in the current study due to its ability to provide a good trade-off between classification accuracy, computational efficiency, and transfer learning abilities. In the proposed framework, ResNet50 is employed not only as an independent benchmark model but also as the convolutional backbone for extracting discriminative local features prior to fusion with Vision Transformer representations. Although ResNet50 effectively captures hierarchical local features, its convolutional receptive fields restrict the modelling of long-range contextual dependencies across the entire leaf image. These limitations motivate the integration of ResNet50 with transformer-based architectures capable of modelling global contextual information, enabling complementary local and global feature learning within the proposed hybrid framework.

3.2 VGG16

One of the earliest deep convolutional neural networks for image classification, known as VGG16, proposed by Simonyan & Zisserman [51], consists of a regular structure with repeating 3×3 convolutional filters and 2×2 max pooling layers. Thirteen convolutional layers and three fully connected layers help in generating hierarchical visual features along with spatial downsampling. A Softmax classifier helps in classifying the final image into the target category. The simplicity of the structure, coupled with pre-trained weights from the ImageNet dataset, makes VGG16 a widely preferred choice for transfer learning for analyzing agricultural images. VGG16 was chosen as a benchmark model for this project because VGG16 is a classic deep convolutional model that has been used widely for transfer learning in the classification of plant diseases, serving as a standard for the effectiveness of the hybrid framework proposed in the research. Several researchers have used the VGG16 model in plant disease detection applications due to its ability to efficiently capture the local lesion features like texture, colour variance, and disease boundaries. On the other hand, VGG16 has about 138 million trainable parameters, leading to a large memory and computational cost during the training and prediction process. Additionally, the deep convolutional structure of this neural network is capable of modeling only the local spatial information without a specific way of incorporating the long-range spatial information from the whole leaf picture. This problem makes the model not effective for separating diseases with the same local information but different global spatial patterns.

3.3 DenseNet121

DenseNet121 [52] introduced by Huang et al. makes use of dense connections between convolutional layers in order to enhance the process of feature propagation and to promote extensive feature reuse. In contrast to traditional CNN architectures which have the input to each layer coming only from the previous layer, all the layers in DenseNet receive the concatenated feature maps produced by all the layers that come before them in the same dense block. This dense connectivity helps in effective gradient propagation and also enables the reuse of features and hence efficient use of network parameters. The network is composed of four dense blocks connected together using transition layers involving convolution and average pooling to progressively reduce the representation size of the features. Because of its feature reuse capability, DenseNet121 has proven successful in applications such as plant disease diagnosis, medical image classification and fine-grained visual recognition which require fine-grained feature discrimination. DenseNet121 was chosen as a baseline model for this work due to the dense feature reuse capability of DenseNet121. Its efficient exploitation of representations learned makes possible a valid comparison with not only traditional convolutional architectures but also the hybrid CNN and Transformer architecture proposed. However, the dense connectivity approach leads to a large increase in memory usage because the feature maps have to be stored during the forward propagation stage. Besides, even though it can effectively propagate features, the DenseNet121 model is still heavily reliant on the convolutional operations, which means that it has little capacity for learning long-term contextual relations between disease symptoms that are separated spatially.

3.4 Vision Transformer

The Vision Transformer (ViT) [53], proposed by Dosovitskiy et al., modifies the transformer model that has been used in natural language processing to classify images. In contrast to the use of convolutional kernels on images, the ViT divides the input image into several patches of a constant size, which are then vectorized and embedded into a sequence of feature vectors learned during training. Positional encodings are

introduced to preserve spatial information prior to applying a series of multiple transformer encoder layers including Multi-Head Self-Attention (MHSA) and Feed Forward Networks (FFNs). The use of self-attention helps the ViT to compute the relationship between all patches in the image at once, enabling the model to identify both global contextual dependencies and relationships between local image patches throughout the image. This characteristic becomes especially useful for the diagnosis of plant diseases as the symptoms often manifest in multiple distant areas of the plant with complex relationships between them. The selection of Vision Transformer as a baseline for this study is motivated by the fact that it presents the basic form of transformers for image classification purposes and serves as a strong benchmark to compare global context feature learning from other CNN-based models. While being more proficient in contextual modelling, Vision Transformer needs larger amounts of annotated data and higher computational power than other CNN architectures. Additionally, the computational complexity of O(N2) for self-attention operations leads to increased usage of GPU memory and higher training time. Therefore, the performance of ViT might be poor while training on small agricultural datasets due to its weak inductive bias and reliance on large annotated data. The combination of Vision Transformer and ResNet50 serves as a solution for these problems through hybridization of global context modelling with local feature extraction.

Table 2 depicts some of the features of the baseline deep learning architectures that have been employed for the purpose of comparative analysis with respect to the proposed ResNet50–Vision Transformer architecture. Some of the features that differ between these deep learning architectures include network architecture, complexity, parameter size, feature representation, and learning methods. In this regard, CNN-based architectures like ResNet50, VGG16, and DenseNet121 are capable of effectively extracting local features in different degrees of computational complexity, while the Vision Transformer is good at capturing global contextual relationship using self-attention mechanism. The combination of these two features leads to the proposed ResNet50–Vision Transformer hybrid architecture.

Table 2 Comparison of the architectural and functional aspects of the baseline deep learning architectures used for moringa leaf disease classification.

Parameter	ResNet50	VGG16	DenseNet121	Vision Transformer (ViT-Base/16)
Architecture	Residual CNN	Deep CNN	Dense CNN	Transformer
Network Depth	50 Layers	16 Layers	121 Layers	12 Encoder Blocks
Input Resolution	224×224	224×224	224×224	224×224
Patch Size	—	—	—	16×16
Trainable Parameters	21.80 M	138.36 M	7.98 M	86.57 M
Model Size	83 MB	528 MB	33 MB	330 MB
GFLOPs	3.66	15.47	2.88	17.58
Feature Dimension	512	512	1024	768
Classifier	Fully Connected + Softmax	Fully Connected + Softmax	Fully Connected + Softmax	MLP Head
Learning Mechanism	Residual Learning	Deep Convolution	Dense Connectivity	Multi-Head Self-Attention

The comparative study of baseline models revealed that each architecture has its unique strengths and drawbacks. Convolutional Neural Networks, like ResNet50, VGG16, and DenseNet121, can successfully extract local features like lesions' texture, color distribution, and edge irregularities; however, they lack capacity to model long-distance relationships. Conversely, Vision Transformer can effectively perform modeling of global context using multi-head self-attention; however, it requires significantly larger sets of annotated data and computational resources. It is clear that both CNNs and transformers have their strengths and weaknesses, which make them not the perfect tools to solve the problem of plant disease classification. Hence, a combination of the two approaches into one is the way to go, as the proposed methodology shows. The details of the approach are provided in the next section.

4. Proposed methodology

4.1 Proposed System Overview

The proposed moringa leaf disease diagnosis framework is developed as a unified multi-stage deep learning pipeline comprising multi-dataset integration, image preprocessing and augmentation, semantic segmentation, hybrid feature learning, disease localization, and explainable artificial intelligence. Fig. 1 represents the moringa leaf disease classification and localization. Initially, images collected from multiple publicly available and field-acquired datasets are integrated to improve data diversity and enhance model generalization. The acquired images subsequently undergo preprocessing and augmentation to improve image quality, reduce noise, and increase dataset variability. The preprocessed images are then segmented using the U-Net architecture to isolate disease-affected leaf regions and suppress irrelevant background information. The segmented images are subsequently processed by the proposed hybrid ResNet50–Vision Transformer model, where ResNet50 extracts discriminative local disease features while the Vision Transformer captures long-range contextual dependencies through multi-head self-attention. The complementary feature representations are fused to perform accurate disease classification. Following classification, YOLOv8 is employed to localize disease-affected regions within the leaf image, while Grad-CAM-based Explainable Artificial Intelligence generates visual explanations highlighting the image regions contributing to the prediction. Contrary to the current methods which are separately applied in segmentation, classification, localization, or explaining, the proposed framework combines these complementary modules into an overall end-to-end system. Therefore, the proposed approach seeks to enhance classification performance, reliability, interpretation, and application of intelligent moringa leaf disease detection in varied agricultural settings.

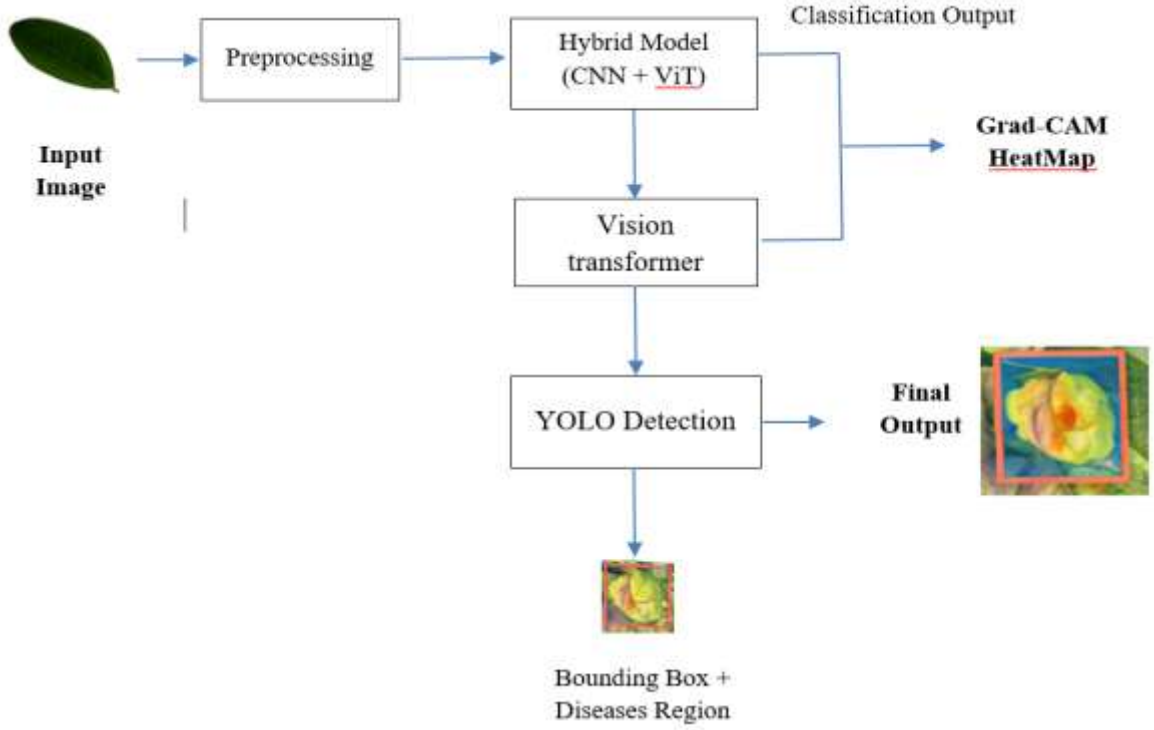


Fig. 1 Hybrid CNN-Vision Transformer (ViT) approach with Grad-CAM and YOLO for interpretable moringa leaf disease classification and localization

4.2 Dataset Preparation

The current work employs a detailed dataset preparation technique to increase the effectiveness of the model. The two main kinds of datasets considered here are (i) the field dataset, which includes real-world images, and (ii) the augmented dataset, which is created using image transformations. This inclusion will help achieve more diversity, prevent overfitting, and increase the efficiency of the model in different environments. In order to formulate a strong and generalized moringa leaf disease diagnosis model, this research employed multiple data sources, including field-captured images and publicly available data that was acquired from Mendeley Data. Combination of datasets with diverse data capturing conditions makes the proposed model more robust to various disease features, lighting conditions, backgrounds, and environments, thus leading to higher generalization ability. Using publicly available datasets further increases the replicability of the research and makes it comparable with other studies.

The field data used were from three moringa farming farms situated within the Thoothukudi District of Tamil Nadu, India, and the collection was carried out between January 2025 and October 2025. The farms that have been surveyed are; farm 1 at Vijayaramapuram (8.4260° N, 77.9189° E), farm 2 at Thisayanvilai

(8.3378° N, 77.8672° E) and farm 3 at Udangudi (8.4293° N, 78.0298° E). These farms are actively involved in moringa farming and have differences in the environment, disease incidence, crop growth stage and farming practices.

Image acquisition was done through the use of the Vivo Y28s 5G phone fitted with a 50 MP RGB camera operating in natural daylight condition without the use of any specialized imaging tool or any lighting. Images were captured at different angles, distances, and perspective in an effort to mimic realistic conditions under which images will be acquired by the farmers. During the fieldwork, both healthy and diseased leaves, with varying stages of disease were purposely selected for inclusion. Also, duplicate and highly corrupted images were discarded after quality assessment. Due to the realistic conditions under which the images were acquired, their original spatial resolution varied with respect to the capture conditions. In an effort to create consistency during the modeling phase, all images were resized to a standard resolution of 224 × 224 pixels using bicubic interpolation.

For enhancing the diversity of the dataset, additional moringa leaf images were acquired from the Mendeley Data repository. The open-source dataset enhances the field-acquired images through the inclusion of more disease samples acquired in different imaging conditions to increase variation in the look of leaves, diseases, backgrounds, and imaging conditions. The combination of the field-acquired and open-source datasets helps reduce the bias of the datasets and allows the proposed framework to learn better feature distributions for deployment in practice. The integrated dataset contains four disease categories: Healthy Leaf, Bacterial Leaf Spot, Cercospora Leaf Spot, and Young Yellow Leaf. All images were manually reviewed and organized according to their respective disease classes. The images were subsequently resized using bicubic interpolation, denoised using Gaussian filtering, and normalized prior to dataset partitioning. After dividing the dataset into training, validation, and test subsets, data augmentation was applied exclusively to the training set to prevent information leakage before semantic segmentation and hybrid CNN–Vision Transformer-based classification. The integration of geographically distributed field-acquired images with a publicly available benchmark dataset provides a biologically representative and diverse image collection, thereby improving the robustness, generalization capability, reproducibility, and practical applicability of the proposed intelligent moringa leaf disease diagnosis framework.

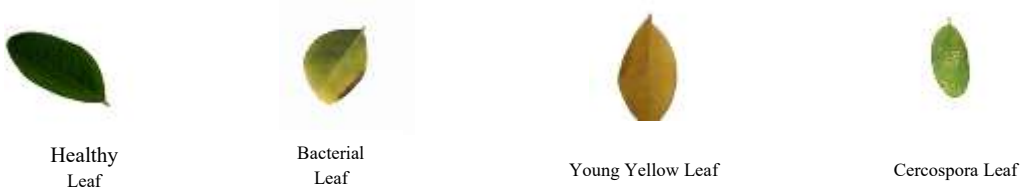


Fig. 2 Sample images obtained from the farms

Table 3 Summary of the integrated moringa oleifera leaf dataset.

Parameter	Description
Crop	Moringa oleifera
Dataset Source	Field-acquired dataset + Publicly available moringa Leaf Disease Dataset (Mendeley Data Repository) [56]
Study Area	Thoothukudi District, Tamil Nadu, India
Survey Period	January 2025 – October 2025
Number of Surveyed Farms	3
Farm 1	Vijayaramapuram (8.4260° N, 77.9189° E)
Farm 2	Thisayanvilai (8.3378° N, 77.8672° E)
Farm 3	Udangudi (8.4293° N, 78.0298° E)
Public Dataset	Mendeley Data repository
Image Acquisition Device	Vivo Y28s 5G Smartphone 50 MP RGB Camera
Image Type	RGB Images
Acquisition Environment	Natural agricultural fields

Lighting Condition	Natural daylight
Disease Classes	Healthy Leaf, Bacterial Leaf Spot, Cercospora Leaf Spot, Young Yellow Leaf
Image Quality Control	Blurred, heavily occluded, and incomplete leaf images were removed
Subsequent Processing	Bicubic resizing, Gaussian filtering, normalization, data augmentation, U-Net segmentation

4.3 Image Pre-processing

Image pre-processing is an important step in the proposed framework since besides improving the image quality, it ensures that the inputs are standardized, thus making the feature extraction process efficient. All the leaf images are first scaled to 224 × 224 pixels through bicubic interpolation, in order to ensure consistency of the input image, yet retaining the disease characteristics of the leaf. In addition, pixel intensity normalization is done to improve numerical stability and increase the speed of model training.

4.3.1 Image Resizing

The process of resizing the image is done to create a standardized input for the hybrid CNN and Vision Transformer (ViT). Since both CNN and ViT models need fixed-size inputs, all moringa leaf images are resized to 224 x 224 pixels prior to feature extraction. Equation 1 represents the original RGB image as

$$I \in R^{H \times W \times C} \quad (1)$$

Where H, W, and C represent the height of the image, width of the image, and number of colour channels, respectively. In the current design, C = 3 since all the leaf images have been collected and stored in the RGB colour space. Every pixel has three values representing the red (R), green (G), and blue (B) channels. It can be noted that the RGB colour space is best suited for plant disease detection because of the difference in colour. Various interpolation techniques can be used for scaling an image. The nearest neighbour interpolation determines the intensity of the new pixel based on its nearest neighbouring pixel, thus producing blocky images with edge discontinuities [54]. Bilinear interpolation produces smooth images, as it takes into account the four nearest neighbours of a pixel, but it might produce blurred edges in the presence of fine structures in the image [54]. On the other hand, bicubic interpolation generates high-quality images because the intensity of each new pixel is determined based on the intensities of the 16 nearest neighbouring pixels (4 x 4 neighbourhood) [55].

In the context of diagnosing diseases on moringa leaves, the use of bicubic interpolation proves to be especially beneficial since it enables the preservation of minute features of disease appearance such as lesions, colour differences, veins and textures. Thus, bicubic interpolation is chosen as the resizing method in the proposed system. Equation 2 represents the resized image as

$$I' = R(I) \quad (2)$$

where R (·) is the resizing method using bicubic interpolation. With this transformation, all images will be standardized in terms of their resolution, making sure that they will have the same resolution of 224 × 224 pixels. Let I (x, y) represent the intensity of the original image at position (x,y). Then, the intensity of the resized image at location (x', y') can be obtained through bicubic interpolation in Equation 3.

$$I'(x', y') = \sum_{m=-1}^{2} \sum_{n=-1}^{2} I(x + m, \quad y + n)\, w(m) w(n) \quad (3)$$

where I (x+m, y+n) refers to the intensity values of the neighbouring pixels, whereas w(m) and w(n) refer to the weights of the bicubic interpolation functions in the x-direction and y-direction, respectively. Using a 4 × 4 neighbourhood, bicubic interpolation produces better-quality images while retaining essential features associated with diseases.

4.3.2 Image Normalization

Normalization of image takes place with the intention of ensuring stability in numbers, increasing the speed of model convergence and improving the efficiency of feature learning process. The reason being that there are wide variations of pixel intensities from one image to another, normalization ensures standardization of the dataset, thus minimizing the effects of lighting changes. Let x be the intensity value of the individual pixel. The normalized intensity value of the pixel is calculated in equation 4 using

$$x_{norm} = \frac{x - \mu}{\sigma} \quad (4)$$

Where μ and σ are the mean and standard deviation of the dataset, respectively. When dealing with RGB images, normalization is done for each colour channel individually. The normalized image can be represented in equation 5 by

$$I'' = \frac{I' - \mu_c}{\sigma_c} \quad (5)$$

Where I′ is the rescaled image, and μ_c and σ_c are the mean and standard deviation of the c^{th} colour channel ($c \in \{R, G, B\}$), respectively. Normalization of the image channels helps in maintaining zero means and unit variances in each colour channel.

4.3.3 Noise Reduction

Images captured in field conditions suffer from several environmental conditions, like changes in illumination, sensor noises, shadow effects, and artifacts in the background. In order to reduce these unnecessary variations but retain disease-associated characteristics, the method of Gaussian filter is used for image smoothing. The filtered image can be obtained in equation 6 by

$$I_{smooth}(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} G(i,j)\, I(x-i, y-j) \quad (6)$$

Where I(x−i,y−j) is the neighbouring pixel intensity value and G(i,j) is the Gaussian kernel. The Gaussian kernel is expressed in equation 7 as:

$$G(i,j) = \frac{1}{2\pi\sigma^2}\, e^{-\frac{i^2+j^2}{2\sigma^2}} \quad (7)$$

Here, σ denotes the amount of smoothing and k represents the kernel size. The advantage of Gaussian filtering is that it reduces noise while maintaining the structural information like leaves, lesions, etc. Hence, the images obtained after pre-processing become useful for further disease detection and localization tasks.

4.3.4 Data Standardization Pipeline

The overall pre-processing scheme combines the processes of resizing, noise removal, and normalization into one scheme. The image transformation represented in equation 8 as follows:

$$I_{final} = N\left(S\left(R(I)\right)\right) \quad (8)$$

Where, R(.) denotes the bicubic interpolation images, S(.) denotes Gaussion Smoothening for noise reduction, N(.) denotes channel wise image normalization

The suggested pre-processing scheme has an important role in improving the performance of the disease detection system. Firstly, it helps to unify the image sizes to provide compatibility for various deep learning models. Secondly, pre-processing leads to the stabilization of the process of training and increases its speed due to normalization and decrease of unwanted variations. Finally, noise removal and background optimization help to filter out unimportant information in order to let the model concentrate on features related to leaf diseases only. This allows extracting such features that would enable the learning of the patterns related to leaf diseases.

4.3.5 Augmented Dataset

Data Augmentation techniques are used to tackle class imbalances, increase the training dataset, and enhance the effectiveness and generalizability of the proposed Deep Learning model. In other words, Data Augmentation does not involve collecting new pictures but creates various examples through transforming images artificially in order to retain the original disease properties of the moringa leaf. In such a way, it helps the model to generalize some important and informative features and prevent overfitting. In order to make the model robust enough for real agricultural conditions, some Data Augmentation techniques are used during the training process. The random rotation technique is used to provide various leaf positions that can be met during capturing an image. The flipping operation both vertically and horizontally is done because of the leaf's symmetrical structure and differences in camera position. To simulate possible illumination changes, such as shadows, low light, and even overexposure, the brightness and contrast adjustment are implemented. Also, scaling and zooming are used in order to simulate various leaf sizes and differences in distances between the camera and leaf surface.

Consider the original image as I and the augmentation operator as A (·). Then the augmented image will be written in equation 9 as

$$I_{augmentation} = A(I) \quad (9)$$

where A (·) is the combination of the set of all the augmentations applied which include rotation, flipping, brightness/contrast manipulation, scaling, and zooming. In other words, the entire augmentation procedure will be written in equation 10 as:

$$A(I) = Z\left(S\left(B\left(F\left(R(I)\right)\right)\right)\right) \quad (10)$$

where R (·), F (·), B (·), S (·), and Z (·) are operations of rotation, flipping, brightness and contrast manipulation, scaling, and zooming, respectively.

Through exposure to various versions of images using the augmented dataset, the learning model improves its capacity to learn invariant features. As a result, the suggested system provides improved classification accuracy, robustness to overfitting and generalizes well when tested on different real-life moringa leaf images acquired under varying environments and conditions.

4.4 Image Segmentation

The symptom spots of the disease usually form a relatively small portion of the leaf's surface, whereas the rest will be healthy leaves with background information. Classification of the image directly without any

pre-processing will have features that do not matter much when it comes to detecting diseases from the images, thus limiting the classification ability of the model in deep learning. Semantic segmentation is used as a step prior to the feature extraction and classification to identify the diseased portion accurately. The method of semantic segmentation is used for identification and segmentation of moringa leaf's disease-infected portions through a neural network which uses U-Net architecture. The semantic segmentation algorithm makes use of classification of pixels in the image, hence providing a clear boundary between the infected leaf portion, the healthy leaf and the image background. Inputs for training the semantic segmentation network include 224 × 224 × 3 dimensional images of the leaf. Outputs of the ground truth masks for the above-mentioned inputs are of 224 × 224 dimensions where each pixel belongs to a particular target class. In order to improve the segmentation network in terms of feature extraction, a pre-trained network is added into the U-Net architecture as part of the encoder using transfer learning.

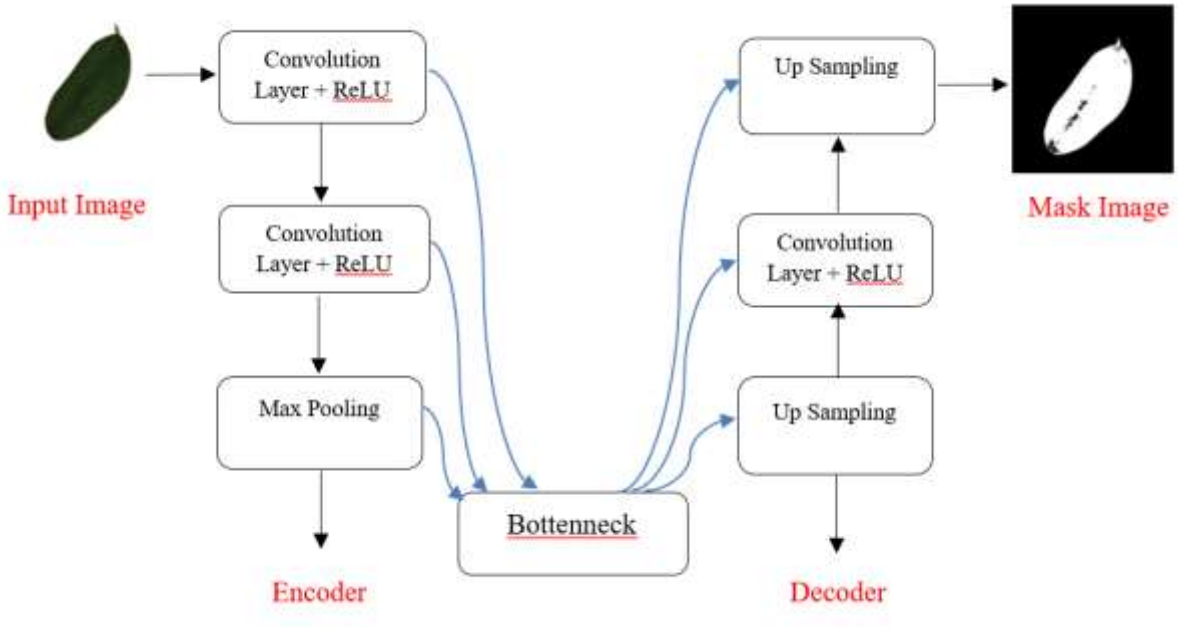


Fig. 3 U-Net architecture for moringa leaf disease semantic segmentation

Fig. 3 shows the overall structure of the proposed U-Net segmentation network. First, the input RGB leaf image undergoes the encoder, which consists of convolutional layers with ReLU activation followed by max-pooling operations. In this process, the encoder not only extracts high-level hierarchical features such as leaf texture, colour information, vein pattern, lesion formation, and diseases but also reduces the spatial resolution of the feature maps. The extracted features are then passed to the bottleneck layer that extracts high-level semantic features necessary for segmenting the diseased areas from the healthy parts of the leaf. In order to retain fine spatial information that can be lost through the process of down sampling, skip connections are formed between the respective layers of the encoder and decoder architecture. Skip connections allow the passing of low-level spatial information from the encoder to the decoder, which enhances the boundary detection capability of the network and helps in precisely locating the disease lesion sites. Ultimately, the network outputs a 224 × 224 pixel-wise segmentation map that assigns one of the three categories to each pixel. The disease regions that have been segmented using the U-Net technique act as the input to the next phase which involves hybrid CNN-Vision Transformer (ViT) classification process. The process helps to eliminate the unnecessary data from the image while highlighting only those regions that are affected by diseases thus making the feature extraction more effective and accurate.

4.5 Hybrid Model Architecture

After pre-processing and U-Net-based semantic segmentation, the segmented moringa leaf images are provided as input to the proposed hybrid deep learning framework. The architecture integrates ResNet50 and Vision Transformer (ViT) to simultaneously extract local spatial features and global contextual information. The fused feature representation is used for disease classification, followed by YOLOv8 for disease localization and Grad-CAM for visual explanation. For successful localization of the local features as well as understanding of global properties of the moringa leaf images, the suggested architecture integrates a Convolutional Neural Network (CNN) and Vision Transformer (ViT). The CNN is known for its ability to extract local spatial features such as texture, color, lesions, and disease-specific features. However, it does not have sufficient capability to understand long-range relations among various parts of the images.

Vision Transformers, on the other hand, use self-attention models to understand global relationships in the images. Thus, by combining these two architectures, the suggested model is able to utilize the advantages of both to perform successful feature extraction and disease classification.

4.3.1 Dataset Integration and Splitting

The data collected from the field and the public Mendeley data were combined to achieve dataset diversity by introducing differences in lighting, background complexity, the extent of the disease, and the imaging conditions. The combined data improves the robustness and generalization capability of the model proposed. Data augmentation was done only on the training data to avoid leaking of information between the training and test datasets. The entire dataset can be expressed in equation 11 as

$$D = D_{field} \cup D_{Mendeley} \cup D_{Augmented} \quad (11)$$

For a reliable development and evaluation of the model, the dataset was split into three parts: 80% training data, 10% validation data, and 10% test data. Training data were used for the estimation of the model parameters, validation data were used for hyper parameter tuning, and the test data served for the performance evaluation on unseen data. Data augmentation helped in enriching the training data by increasing the variety and balance of the classes; however, the validation and test datasets were not changed at all.

Table 4 shows the data distribution for the moringa leaf disease dataset employed in this work. The dataset was formed through the integration of 2,000 images from the field, 1,660 Mendeley images, and 5,272 augmented images, which led to a total of 8,812 images in the dataset. The dataset contains four categories of leaves, namely Healthy, Bacterial Leaf Spot, Cercospora Leaf Spot, and Young Yellow Leaf. There are 2,312, 2,312, 2,088, and 2,220 images for each of these categories, respectively.

Table 4 Distribution of integrated moringa leaf disease image dataset

Source	Healthy	Bacterial	Cercospora	Young Yellow	Total
Field Dataset	500	500	500	500	2000
Mendeley Dataset	415	415	415	415	1660
Augmented Dataset	1397	1397	1173	1305	5272
Final Dataset	2312	2312	2088	2220	8812

Table 5 shows how the distribution of the final dataset is carried out in training, validation, and test datasets. In line with the 80:10:10 data splitting approach, 7,050 images (80%) were assigned to the training set, 881 images (10%) to the validation set, and 881 images (10%) to the test set. The training set was employed for learning feature representation and parameter optimization, while the validation set helped with tuning hyper parameters and selecting a suitable model. The test set, which was kept independent of the other sets, was used for the evaluation of the proposed framework's performance.

Table 5 Training, validation, and test dataset split for model development

S.No	Type of Leaves	Train Images	Validation Images	Test Images	Total Images
1	Healthy Leaf	1850	231	231	2312
2	Bacterial Leaf Spot	1850	231	231	2312
3	Cercospora Leaf spot	1670	209	209	2088
4	Young Yellow Leaf	1776	222	222	2220
Total Images		7050 (80%)	881 (10%)	881 (10%)	8812 (100%)

4.3.2 CNN Feature Extraction

ResNet50 acts as the first step of feature extraction in the hybrid approach that is introduced. For an input segmented moringa leaf image I, the network passes the image through multiple convolutional layers, residual blocks, and the ReLU activation function in order to extract hierarchical spatial features. The extracted features range from lower-level features like edges, corners, and color gradients to higher-level

features like disease-specific characteristics such as lesions, spots, and infected tissue. The extracted feature representation can be written in equation 12 as

$$F_{cnn} = f_{cnn}(I) \quad (12)$$

where $F_{cnn} \in R^{d_1}$ represents the higher-level spatial feature representation, while fcnn(·) denotes the ResNet50 feature extraction model. The residual connections help learn deeper features by addressing the issue of vanishing gradients, allowing the network to learn the complicated disease characteristics from the moringa leaves images.

4.3.3 Vision Transformer (ViT)

Although CNNs are concerned mostly with capturing local features, the Vision Transformer model is used for acquiring the global context of the leaf images. The resized image of 224 × 224 pixels is divided into 16 × 16 non-overlapping patches, resulting in 196 image patches, each represented as a visual token. If the number of patches formed is N, then image patches represented in equation 13 as:

$$x_p \in R^{N\times(P^2.C)} \quad (13)$$

where P stands for the size of patches and C stands for the number of channels in an image. These patches are then converted to embedding vectors by applying linear projection. The concept of positional embedding's is introduced to keep the position of the image patches intact represented in equation 14.

$$z_0 = \left[x_{class};\ x_p^1 E;\ x_p^2 E;\ \ldots . x_p^N E\right] + E_{pos} \quad (14)$$

This encoded sequence then goes through several transformer encoder layers. The vision transformer, unlike convolution operations, processes all image patches at once and is able to learn dependencies between distant regions of an image. Such a feature is very helpful in the detection of disease symptoms present in different parts of the leaf. The main module of Vision Transformer architecture is multi-headed self-attention, which allows the vision transformer to know the importance of the particular image patch compared to other patches. Self-attention is calculated using equation 15 as:

$$Attention\ (Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (15)$$

where Q, K, and V represent the query, key, and value matrices, respectively, while dk denotes the dimension of the key vector. This helps the model to concentrate on the important sections of the image while making predictions. For instance, in case of the appearance of the symptoms of a disease on multiple disconnected parts of the leaf, then the attention process can easily draw a relationship between the sections. The final output from the transformer is denoted in equation 16 as:

$$F_{vit} = f_{vit}(I) \quad (16)$$

where $F_{vit} \in R^{d2}$ captures global context dependencies and semantic information, thus complementing the local features extracted by the CNN branch.

4.3.4 Feature Fusion

In order to make use of the benefits of both the architectural designs, the features from CNN and Vision Transformer are merged using the feature fusion technique. The CNN network provides detailed local features of the diseased textures and lesions, whereas the Vision Transformer provides global context understanding about the whole leaf structure. The fusion operation is defined in equation 17 as:

$$F_{fusion} = [F_{cnn} \,||\, F_{vit}] \quad (17)$$

where || is the notation for feature concatenation. Such a fused feature vector provides a more informative and discriminative representation compared to the individual feature sets. Through such feature fusion of local and global features, the model gains the ability to differentiate between diseases that have highly similar symptoms and even when they are in difficult environmental settings.

4.3.5 Classification Layer

Once the features have been fused, they will then be passed to one or more fully-connected layers that will classify the disease. This layer creates complicated decision planes based on the combination of the local and global features obtained from the input image. The classification can be represented in equation 18 by:

$$y = \sigma\left(WF_{fusion} + b\right) \quad (18)$$

where W and b denote the learnable weights and biases, respectively, while σ denotes the Softmax activation function. The Softmax layer transforms the output to a probability distribution of all disease classes. The disease class predicted can be obtained from equation 19.

$$\hat{y} = arg\ max(y) \quad (19)$$

where ŷ denotes the class with the maximum probability. With the help of the aforementioned classification approach, the model can classify healthy leaves and various diseases using the features extracted from the input image.

4.3.6 Loss Function

The network parameters are optimized using the Adam optimizer with categorical cross-entropy loss. The categorical cross-entropy loss function is used for the optimization of the model parameters during the training process. Mathematically, loss can be expressed in equation 20 as follows:

$$L = -\sum_{i=1}^{C} y_i \, log(\hat{y}_i) \qquad (20)$$

where C stands for the total number of disease classes, y_i is the actual label, and $\hat{y}_i$ *is* the prediction probability for class i. The main goal of the training process is to minimize the loss value by continuously optimizing the network parameters. In this way, during the training phase, the model gains better feature representation capacity which ultimately leads towards better classification performance and robustness when applied to real-life problems related to moringa leaf diseases.

Several strengths can be attributed to the suggested hybrid CNN–Vision Transformer (ViT) approach to the classification of moringa leaves in terms of their diseases. Namely, the combination of the CNN part responsible for local features extraction and the transformer part with attention mechanism enables the algorithm to identify various types of leaf diseases, including those related to leaf texture, colors, disease-specific shapes of the affected area, and other aspects. At the same time, the integration of local and global feature representations helps enhance the accuracy and robustness of the system in comparison with separate CNN and transformer architectures. In addition, the developed approach allows the system to work with complex disease symptoms varying in terms of sizes and shapes, which cannot be done using a single model type. Overall, the hybrid architecture solves the existing limitations of separate models and makes feature learning more efficient.

4.6 Disease localization using YOLOv8

In order to improve the practical utility of the suggested moringa leaf disease detection system, disease localization is carried out using the algorithm known as You Only Look Once Version 8 (YOLOv8). This method has been chosen due to its efficiency and superior performance as compared to previous YOLO architectures, which include high localization precision, fast computation time, and reduced computational complexity. It should be noted that YOLO is distinct from image classification methods in the way that it can do localization and classification at the same time. Given an input image $I \in R^{H \times W \times 3}$, The YOLOv8 algorithm processes the input image through its backbone, neck, and detection head to generate a set of predictions which is represented in equation 21.

$$Y = \{(b_i, c_i, p_i)\}_{i=1}^{N} \qquad (21)$$

where $b_i = (x_i, y_i, w_i, h_i)$ represents the bounding box coordinates, c_i denotes the predicted disease class, and p_i represents the confidence score associated with each detection. This design allows YOLO to recognize several diseased areas at once without sacrificing its computational efficiency. The coordinates and dimensions of the bounding box are computed using equation 22, 23, 24, and 25 as follows:

$$x = \sigma t_x + c_x \qquad (22)$$

$$y = \sigma t_y + c_y \qquad (23)$$

$$w = p_w e^{t_w} \qquad (24)$$

$$h = p_h e^{t_h} \qquad (25)$$

where t_x, t_y, t_w, and t_h are represent the predicted offsets obtained from the network, c_x and c_y are the coordinates of the grid cell, and p_w, p_h are anchor box dimensions. The use of such parameterization ensures that the model is able to detect bounding boxes of different sizes and shapes; therefore, the algorithm is able to accurately locate the disease spots irrespective of their location or scale within the leaf. Once the bounding boxes are identified, the YOLO model calculates a confidence score of each detection. The confidence score (p) measures the presence of the object and the precision of the bounding box. It is defined in equation 26:

$$p = P(object) \times IOU_{pred}^{truth} \qquad (26)$$

where P(object) refers to the probability that the bounding box contains a region with diseases. IOU is Intersection over Union which calculates the overlap between the predicted and actual bounding box. Confidence score is high when the model is more certain about the detection. Also, the probability of the class is predicted by the Softmax function represented in equation 27.

$$P(class_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \qquad (27)$$

where z_i refers to the score assigned to class i and C denotes the number of diseases present in the class. It helps classify any detected area with the correct disease. In order to improve detection, YOLO uses an aggregate loss function that tries to minimize errors related to localization, confidence, and classification. The loss function can be written in equation 28 as follows:

$$L = \lambda_{coord} L_{loc} + L_{conf} + L_{cls} \qquad (28)$$

where L_{loc} stands for the localization loss from bounding box regression, L_{conf} for the confidence score loss, and L_{cls} for the classification loss. The weight λ_{coord} plays a significant role in balancing the impact of the localization loss during training. The model trained using the two parameters mentioned above will be able to localize the diseased regions and classify the diseases into their categories accurately. In object detection, the model can predict several bounding boxes for the same diseased region. In order to remove the unnecessary boxes and retain only the correct bounding box, the model makes use of non-maximum suppression. This suppresses all boxes with an overlap of more than the threshold value. This process is represented mathematically as follows:

$$NMS(B) = max_i\,(p_i) \qquad IOU < \theta \qquad (29)$$

Where p_i is the confidence score and θ is the overlap threshold value. This method enhances the performance of the model by retaining only the bounding boxes with high confidence.

There are many advantages associated with the use of YOLO in disease localization in moringa leaves. As this algorithm is capable of detecting diseases in real time, it can be easily implemented in the field and even in mobile devices. With great localization accuracy due to precise identification of disease areas and generation of accurate bounding boxes, the proposed system is convenient to use. In addition, single-stage detection saves computational power and decreases inference time when compared to traditional two-stage systems. Moreover, it is important to mention that performing both tasks at once increases usability of the proposed approach. This way, farmers and other agricultural workers can efficiently localize the disease and take necessary actions.

4.7 Explainable AI using Grad-CAM

The Gradient-weighted Class Activation Mapping (Grad-CAM) method is used to achieve the aim of having Explainable Artificial Intelligence (XAI) in order to make the proposed moringa leaf disease prediction system more transparent and interpretable. The Grad-CAM method is used during the inference step but not during the training of the model. It provides visual explanations by emphasizing the image parts that are crucial for determining the class of disease.

Let A^k represent the k^{th} feature map in the final convolutional layer and y^c denote the output score corresponding to class c. The importance weight for each feature map is calculated using equation 30 as:

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k} \qquad (30)$$

where α_k^c represents the importance weight associated with feature map k, A_{ij}^k denotes the activation value at location (i,j), and Z is a normalization factor corresponding to the spatial dimensions of the feature map. The heatmap is calculated using equation 31 as:

$$L_{Grad-CAM}^c = ReLU\left(\sum_k \alpha_k^c A^k\right) \qquad (31)$$

Where $\boldsymbol{L_{Grad-CAM}^c}$ represents the localization map for class c. The Rectified Linear Unit (ReLU) function is applied to retain only the positive contributions, ensuring that the visualization highlights image regions that positively influence the model's prediction. The visualization process can be expressed using equation 32 as:

$$I_{cam} = I + \lambda . L_{Grad-CAM} \qquad (32)$$

where I denote the original image, $\boldsymbol{L_{Grad-CAM}}$ represents the generated heatmap, and λ controls the intensity of the overlay. The visualization technique thus obtained shows the regions which contribute to the decision making process of the model, thus helping researchers and agricultural experts to confirm that the network is considering biologically significant regions of diseases. Grad-CAM makes the proposed framework more interpretable by showing disease-specific regions that are contributing towards the classification process. This method also helps in detecting any misclassifications, thus ensuring reliability and usability of the proposed moringa leaf disease detection system.

4.8 Computational Complexity Analysis

Besides the efficiency in classifying diseases in crops, computational efficiency is very critical when implementing disease diagnosis models on edge devices and mobile agriculture platforms. In that regard, computational complexity of the suggested ResNet50-Vision Transformer (ViT) architecture is analyzed and compared to other commonly employed models such as ResNet50, VGG16, DenseNet121, and Vision Transformer (ViT). This analysis takes into consideration the computation cost, memory consumption, training time, inference time, and the size of the model to establish the most suitable model for real-time agriculture use cases.

Computational complexity of any deep learning model depends on the number of trainable parameters, the number of FLOPs (floating-point operations), GPU memory consumption, training time, and inference time. Computational Complexity expressed in equation 33 as

$$\boldsymbol{C = f(P, F, M, T, I)} \qquad \boldsymbol{(33)}$$

Where, P denotes the number of trainable parameters, F represents the floating-point operations (GFLOPs), M denotes GPU memory utilization, T represents training time, I denote inference time. The computational cost of CNN is estimated using equation 34 as

$$F_{CNN} = 2 \times H \times W \times C_{in} \times C_{out} \times K^2 \quad (34)$$

where H, W are the spatial dimensions, C in and C out denote input and output channels, K is the convolution kernel size. For the Vision Transformer, the computational cost of convolutional layers is estimated using equation 35 as

$$F_{ViT} = O(N^2 d) \quad (35)$$

where N denotes the number of image patches, d represents the embedding dimension. Unlike convolutional neural networks, the computational cost of the transformer increases quadratically with the number of image patches because every patch attends to every other patch. The proposed hybrid framework combines ResNet50 and Vision Transformer, enabling simultaneous extraction of local and global feature representations. Consequently, its computational complexity can be approximated using equation 36 as

$$F_{Hybrid} = F_{ResNet50} + F_{ViT} + F_{Fusion} \quad (36)$$

where F_{Fusion} denotes the computational cost associated with feature fusion and classification.

Table 6 shows the comparison of the computational complexity of the proposed method with the existing deep learning methods. Though the proposed ResNet50-ViT has high computational complexity compared to the existing CNN-based models, the proposed model obtains the highest classification accuracy of Top-1 (99.68%), with competitive computational efficiency and low inference time (14.2 ms per image). From the above discussion, we can see that the proposed approach obtains the right balance between computational complexity and classification accuracy.

Table 6 Computational complexity comparison of existing and proposed models

Metric	ResNet50	VGG16	DenseNet121	Vision Transformer (ViT)	Proposed ResNet50 + ViT
Trainable Parameters (Millions)	21.80	138.36	7.98	86.57	110.12
FLOPs (GFLOPs)	3.66	15.47	2.88	17.58	21.30
GPU Memory Usage (GB)	1.65	4.28	1.22	3.10	3.95
Training Time / Epoch (s)	42	91	38	74	69
Inference Time / Image (ms)	13.8	18.9	12.4	16.5	14.2
FPS	80.6	48.5	92.4	65.8	68.5
Model Size (MB)	83	528	33	330	420
GPU Utilization (%)	71	93	68	86	88
Power Consumption (W)	142	221	136	205	198
Top-1 Classification Accuracy (%)	96.82	97.41	98.26	98.74	**99.68**

5. Experimental results and performance evaluation

5.1. Experimental Setup

The proposed hybrid deep learning framework was implemented using Python 3.11 in the Google Colaboratory (Google Colab) cloud computing environment. The model was developed using the PyTorch deep learning framework, with TorchVision for pre-trained model implementation and image transformations. Additional libraries including OpenCV, NumPy, Pandas, Scikit-learn, and Matplotlib were employed for image preprocessing, data augmentation, numerical computation, performance evaluation, and visualization. The experiments were accelerated using the NVIDIA Tesla T4 GPU (16 GB VRAM) available in Google Colab with CUDA support. The proposed framework comprises U-Net for semantic segmentation, a hybrid ResNet50–Vision Transformer (ViT-Base/16) for disease classification, YOLOv8 for disease localization, and Grad-CAM for model interpretability. The dimensions of all input RGB images were scaled to 224 × 224 × 3, and that of their segmentation masks to 224 × 224. The hybrid classification model was trained with the Adam optimizer and an initial learning rate of 1×10^{-4}, a batch size of 32, and 12 training epochs. The adaptive adjustment of learning rates was done through the use of Reduce LROnPlateau scheduler with a reduction factor of 0.2 and a patience of 5 epochs, whereas Early Stopping with a patience of 10 epochs was used to avoid overfitting. The classification of diseases was based on Cross

Entropy Loss, while the optimization of U-Net for the task of segmentation was achieved through Dice Loss in addition to BCE Loss. Accuracy, precision, recall, F1-score, specificity, AUC score, confusion matrix, inference time, and computational complexity were considered as evaluation metrics of the proposed framework. Table 7 represents the requirement of proposed system design.

Table 7 Experimental setup of the proposed hybrid framework

Parameter	Value
Programming Language	Python 3.11
Development Platform	Google Colaboratory (Google Colab)
Deep Learning Framework	PyTorch
GPU	NVIDIA Tesla T4 (16 GB VRAM)
CUDA Support	Enabled
Image Size	224 × 224 × 3
Segmentation Mask Size	224 × 224
Backbone CNN	ResNet50
Transformer	ViT-Base/16
Segmentation Model	U-Net
Object Detection	YOLOv8
Explainability	Grad-CAM
Optimizer	Adam
Initial Learning Rate	1×10^{-4}
Batch Size	32
Number of Epochs	12
Learning Rate Scheduler	ReduceLROnPlateau
Early Stopping	Patience = 10
Classification Loss	Cross-Entropy Loss
Segmentation Loss	Dice Loss + Binary Cross-Entropy
Evaluation Metrics	Accuracy, Precision, Recall, F1-Score, Specificity, AUC, Confusion Matrix, Inference Time, Computational Complexity

5.2. Pre-processing Results

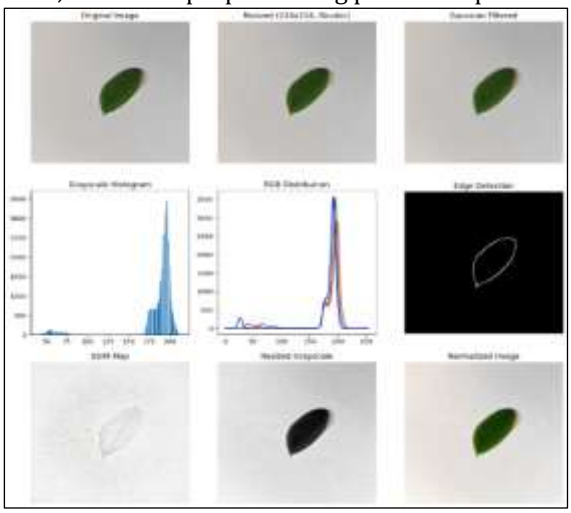
For the purpose of analysing the performance of the suggested image pre-processing approach, the resulting pre-processed images were tested based on several image quality measures, such as Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Signal to Noise Ratio (SNR), entropy, average pixel value, and standard deviation. All these measures help analyze pixel-wise distortion, structure preservation, noise reduction, and information loss caused by image resizing, Gaussian blurring, and normalization operations. Table 8 represents the image analysis results for various types of moringa leaves

Table 8 Quantitative image quality evaluation of pre-processing techniques for moringa leaf images.

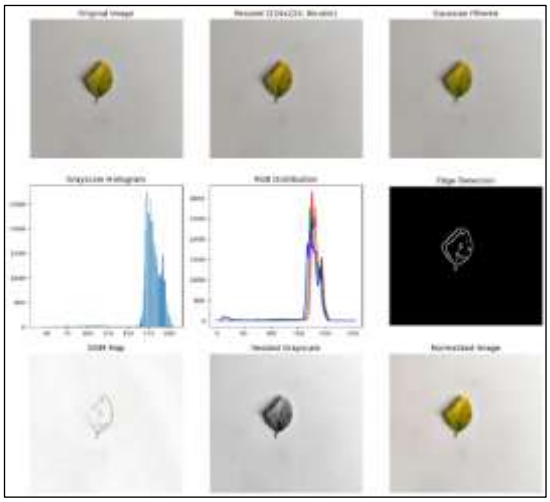
Type of Images	Pre-processing Step	Image Quality Evaluation Metrics						
		MSE	PSNR (dB)	SSIM	SNR (dB)	Entropy	Mean	Std Dev
Healthy Leaves	Bicubic Resizing (224×224)	8.8095	38.6813	0.9377	35.9814	5.3619	183.7816	33.0280
	Gaussian Filtering	13.1708	36.9347	0.9126	34.2352	5.3135	183.9059	32.6636
	Normalization	940.3081	18.3981	0.9763	15.6941	5.3135	212.8007	42.9267
Bacterial Leaf Spot	Bicubic Resizing (224×224)	4.6937	41.4156	0.9676	38.3050	5.4278	176.7558	21.8675
	Gaussian Filtering	7.0250	39.6643	0.9665	36.5538	5.4134	176.8803	21.5572
	Normalization	1261.5286	17.1218	0.9801	14.0087	5.4134	211.9450	27.2076

Cercospora Leaf Spot	Bicubic Resizing (224×224)	19.9171	35.1385	0.8631	32.0275	5.8077	176.4468	23.9632
	Gaussian Filtering	17.5080	35.6984	0.8704	32.5874	5.7796	176.5743	23.5583
	Normalization	836.2384	18.9075	0.9811	15.7919	5.7796	204.2426	31.9634
Yellow Leaf	Bicubic Resizing (224×224)	14.2016	36.6074	0.8878	31.9940	5.5034	148.8339	16.5243
	Gaussian Filtering	11.6739	37.4587	0.8897	32.8441	5.4693	148.9587	16.1702
	Normalization	3503.4962	12.6858	0.9347	8.0672	5.4693	207.3919	25.6045

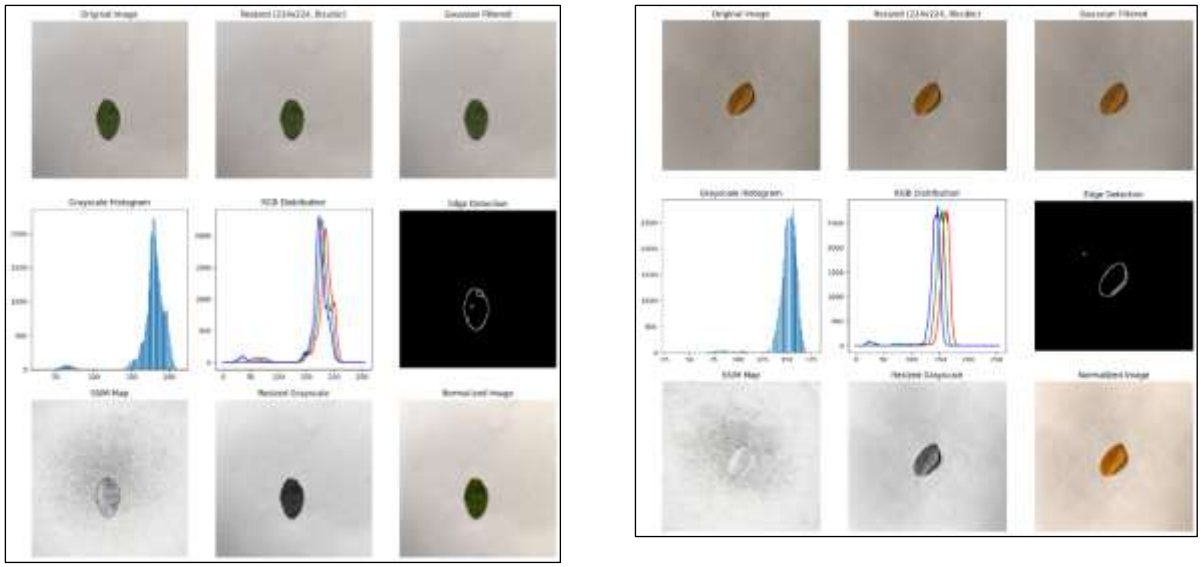
Fig. 4 represents the image pre-processing results of some moringa leaves diseases categories. In Healthy Leaf category, the original leaf image with 3072 x 3072 x 3 size is downscaled to 224 x 224 x 3 using bicubic interpolation, then reconstructed back to its original size to assess the quality. Statistical properties of the reconstructed image are similar to the original image, with a mean value of 183.898 and 183.899, and a standard deviation of 33.210 and 33.087, respectively. In addition, the PSNR and SSIM of the reconstructed image are 39.853 dB and 0.94145, respectively, implying that the resizing operation kept the structural properties of the original leaf image. Healthy Leaf images using bicubic interpolation gives MSE = 8.8095, PSNR = 38.6813 dB, SSIM = 0.9377 and Gaussian Filter produces MSE = 13.1708, PSNR = 36.9347 dB, SSIM = 0.9126, but after normalization, SSIM value increased to 0.9763. For Bacterial Leaf Spot images, bicubic resizing provided the minimum MSE (4.6937) along with the maximum PSNR (41.4156 dB) and SNR (38.3050 dB). Gaussian filter had similar SSIM values, but normalization led to higher MSE values (SSIM = 0.9801). In the case of Cercospora Leaf Spot images, Gaussian filter showed the following results: MSE = 17.5080, PSNR = 35.6984 dB, SSIM = 0.8704, and SNR = 32.5874 dB. The same was true for Young Yellow Leaf images with the following parameters: MSE = 11.6739, PSNR = 37.4587 dB, SSIM = 0.8897, and SNR = 32.8441 dB. Normalization kept the image structure similar but changed the intensity distribution of pixels. Thus, the entire preprocessing process kept the structural information but normalized.



b. Healthy Leaf



a. Bacterial Leaf Spot



c. Cercospora Leaf d. Young Yellow Leaf

Fig. 4 Results of image pre-processing of representative moringa leaf diseases.

The Figure 4 shows the representative images of the original, preprocessed, and augmented moringa leaf images for the four moringa leaf disease categories, which are Healthy Leaf, Bacterial Leaf Spot, Cercospora Leaf Spot, and Young Yellow Leaf. The original images that have been captured in real-field environments underwent preprocessing steps of resizing, normalization, and denoising. Data augmentation processes including rotating were performed on the original images in order to create more images while maintaining the inherent properties of the disease. The image preprocessing process ensures the consistency of the image by eliminating the effects of backgrounds and illumination, while the data augmentation increases the data diversity.

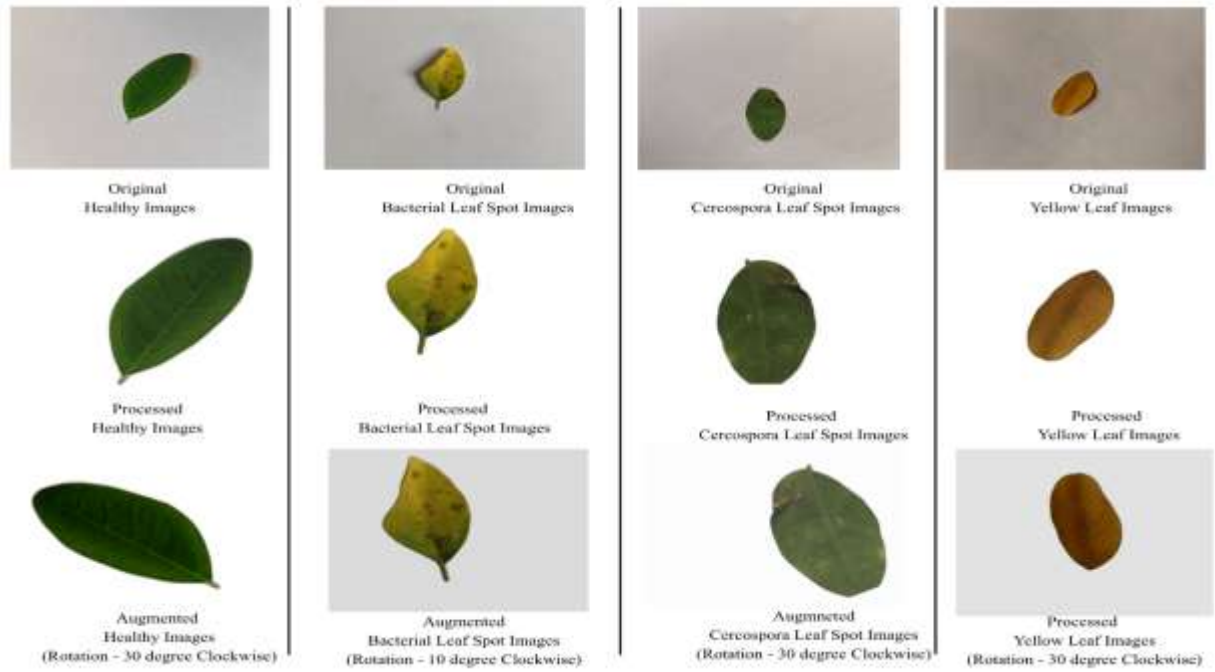


Figure 5. Illustration of representative original, preprocessed, and augmented moringa leaf images from each disease category.

5.3 Image Segmentation

To evaluate the effectiveness of the proposed U-Net model, semantic segmentation was performed to identify disease-infected regions prior to classification. The segmentation network was trained for four epochs, and its performance was monitored using segmentation accuracy and training loss. The objective of this stage was to accurately separate diseased leaf regions from the background, thereby providing meaningful inputs for the subsequent hybrid ResNet50–Vision Transformer classification model. Table 5.3 presents the segmentation performance obtained during training.

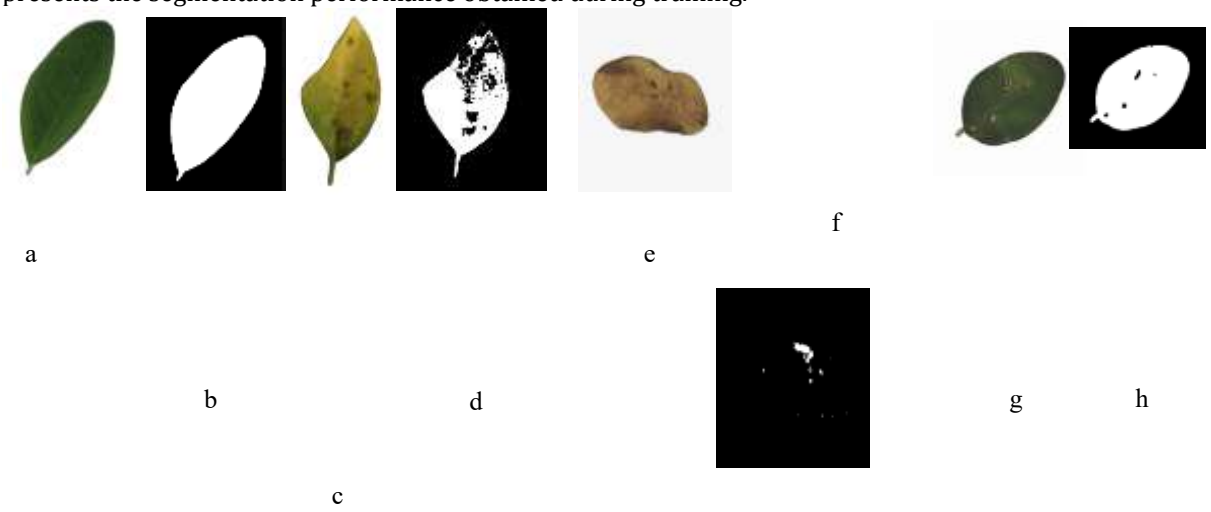


Figure 6(a). Health Leaf, 6(b). Mask image of Healthy Leaf, 6(c). Bacterial leaf, 6(d) Mask image of bacterial Leaf, 6(e). Young Yellow Leaf, 6(f). Mask image of young yellow leaf, 6(g). Cercospora Leaf, 6(h). Mask image of Cercospora leaf.

Table 9. Training performance of the proposed U-Net model for moringa leaf image segmentation.

Image Segmentation (U-Net) Results21				
Epoch	1	2	3	4
Accuracy	0.9782	0.9811	0.9876	0.9923
Loss	0.0377	0.019	0.012	0.008

The segmentation model demonstrated consistent improvement throughout the training process which is shown in table 9. The segmentation accuracy increased from 97.82% in the first epoch to 99.23% in the fourth epoch, while the training loss decreased from 0.0377 to 0.0080. The continuous increase in accuracy together with the corresponding reduction in loss indicates stable convergence of the U-Net model during training. The obtained segmentation results demonstrate that the proposed U-Net architecture effectively learned the pixel-level representation of moringa leaf images and accurately separated the disease-affected regions from the background. These segmented masks were then employed as input data for the proposed hybrid model based on ResNet50 and Vision Transformer, which allowed for the classification network to concentrate on disease-related regions and ignore background information.

5.4 Performance Comparison of Existing and Proposed methods

The classification performance of the proposed ResNet50–Vision Transformer (ResNet50–ViT) framework was quantitatively evaluated using standard multi-class performance metrics, including accuracy, precision, recall, F1-score, specificity, Area Under the Receiver Operating Characteristic Curve (AUC), and inference time. These metrics collectively provide a comprehensive assessment of prediction accuracy, classification reliability, discriminative capability, and computational efficiency. The performance of the proposed framework is evaluated using the following standard classification metrics.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (37)$$

$$Precision = \frac{TP}{TP + FP} \times 100 \quad (38)$$

$$Recall = \frac{TP}{TP + FN} \times 100 \quad (39)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100 \quad (40)$$

$$Specificity = \frac{TN}{TN + FP} \times 100 \quad (41)$$

$$Area\ Under\ the\ ROC\ Curve(AUC) = \int_0^1 TPR(FPR)\, d(FPR) \quad (42)$$

$$TPR = \frac{TP}{TP + FN} \quad (43)$$

$$FPR = \frac{FP}{FP + TN} \quad (44)$$

$$Inference\ Time = \frac{Total\ Prediction\ Time}{Number\ of\ Test\ Images} \quad (45)$$

where TP, TN, FP, and FN denote the number of true positive, true negative, false positive, and false negative predictions, respectively. These quantities are used to compute the standard performance metrics for evaluating the proposed classification model. Table 5.4 summarizes the quantitative comparison between the proposed framework and four widely adopted deep learning models using the aforementioned evaluation metrics.

Table 10. Quantitative comparison of the proposed ResNet50–Vision Transformer framework with existing deep learning models for moringa leaf disease classification.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)	AUC	Inference Time (ms)
ResNet50	96.82	96.35	96.28	96.31	98.42	0.981	13.8
VGG16	97.41	97.12	97.03	97.07	98.75	0.986	18.9
DenseNet121	98.26	98.05	97.92	97.98	99.10	0.992	12.4
Vision Transformer	98.74	98.61	98.53	98.57	99.32	0.995	16.5
Proposed ResNet50 + ViT	**99.68**	**99.54**	**99.47**	**99.50**	**99.81**	**0.998**	**14.2**

The quantitative results presented in Table 10 demonstrate that the proposed ResNet50–Vision Transformer (ResNet50–ViT) framework achieved the highest classification performance among all evaluated models. The proposed framework achieved an overall classification accuracy of 99.68%, together with a precision of 99.54%, recall of 99.47%, F1-score of 99.50%, specificity of 99.81%, and an AUC of 0.998, indicating consistently high predictive performance across all evaluation criteria. Compared with the baseline models, the proposed framework improved the overall classification accuracy by 2.86%, 2.27%, 1.42%, and 0.94% over ResNet50, VGG16, DenseNet121, and Vision Transformer, respectively. As for all the tested models, DenseNet121 achieved the fastest inference speed (12.4 ms), but the developed model needed only 14.2 ms per one image.

5.5 Confusion Matrix

The confusion matrix provides a detailed evaluation of the classification performance of the proposed ResNet50–Vision Transformer framework by comparing the predicted labels with the corresponding ground truth classes. It enables the identification of correctly classified samples and misclassification patterns across the four moringa leaf categories, namely Healthy Leaf, Bacterial Leaf Spot, Cercospora Leaf Spot, and Young Yellow Leaf. Table 5.5 presents the confusion matrix obtained using the test dataset.

Figure 7. Confusion matrix of the proposed ResNet50–Vision Transformer framework showing the classification results for Healthy Leaf, Bacterial Leaf Spot, Cercospora Leaf Spot, and Young Yellow Leaf images.

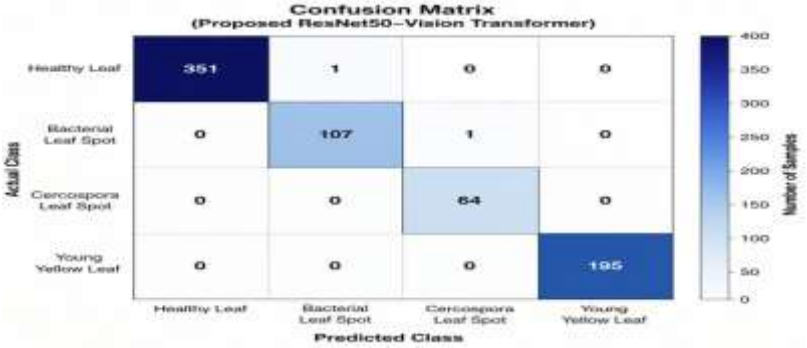


Figure 7. Confusion matrix of the proposed hybrid ResNet50–Vision Transformer model on the test dataset.

The quantitative results shown in figure 7 indicate that the proposed framework correctly classified 351 of 352 Healthy Leaf images, with only one sample misclassified as Bacterial Leaf Spot. Similarly, 107 of 108 Bacterial Leaf Spot images were correctly identified, while one sample was classified as Cercospora Leaf Spot. All 64 Both the Cercospora Leaf Spot and 195 Young Yellow Leaf categories were identified accurately with no instances of misclassification. In total, there were just two cases of misclassification among the 719 test images, confirming that the proposed method is highly accurate in classifying images. The confusion matrix clearly shows a strong presence of diagonal values with very few non-diagonal entries, showing that the proposed method successfully differentiates between the four disease categories, while at the same time keeping the rate of misclassification very low. The cases of misclassification were found only between disease categories which look alike, while no misclassification was found between the Healthy Leaf and Yellow Leaf categories or between the Cercospora Leaf Spot and Yellow Leaf categories.

5.6 Training Accuracy and Loss for Hybrid Model

The performance of the proposed ResNet50-Vision Transformer framework in terms of learning, convergence, and generalization capability was investigated over 12 training epochs. The framework was trained with the Adam optimizer having a fixed learning rate of 0.0001. Accuracy and loss values of training and validation were captured at the end of each epoch to trace the optimization process. Table 11 presents the training history of the proposed framework.

Table 11. Training history of the proposed ResNet50–Vision Transformer framework

No. of Epochs	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Learning Rate
1	0.8818	0.5189	0.9656	0.1211	0.0001
2	0.9825	0.0597	0.9752	0.0832	0.0001
3	0.9878	0.0403	0.9800	0.0693	0.0001
4	0.9912	0.0312	0.9793	0.0635	0.0001
5	0.9918	0.0262	0.9842	0.0548	0.0001
6	0.9927	0.0228	0.9842	0.0531	0.0001
7	0.9935	0.0195	0.9828	0.0547	0.0001
8	0.9941	0.0164	0.9821	0.0548	0.0001
9	0.9951	0.0156	0.9759	0.0622	0.0001
10	0.9955	0.0141	0.9738	0.0674	0.0001
11	0.9965	0.0114	0.9800	0.0584	0.0001
12	0.9968	0.0105	0.9787	0.0588	0.0001

From the above experiment, there is an evident increase in the training performance in terms of accuracy and reduction in the loss values from the first to the last epoch of training. For instance, the training accuracy improved from 88.18% in the first epoch to 99.68% in the twelfth epoch. The training loss dropped significantly from 0.5189 in the first epoch to 0.0105 in the last epoch of training. An increase in the accuracy followed by converging accuracy was experienced in the initial and the later training phases respectively. In addition, the validation accuracy was consistently high throughout the training period; it rose from 96.56% in the first epoch of training to the peak value of 98.42% during training to finally settle at 97.87% in the final epoch of training. Validation loss reduced significantly at the initial phase of training and afterwards remained consistently in a narrower range of 0.0531 to 0.0674 in the later training epochs. Learning rate was consistently 0.0001 throughout the training epochs. It is clear that there is consistency in the training and validation performance through convergence with increasing accuracy and reducing loss values in successive training epochs.

5.7 Comparison of Training Accuracy

In order to assess the convergence properties of the proposed framework, the training accuracy was compared with four well-known deep learning models, that is, ResNet50, VGG16, DenseNet121, and Vision Transformer, for 12 training epochs. The following table shows the learning process and the convergence of the proposed framework with respect to the baseline models. Table 12 Comparison of training accuracy (%) between the proposed ResNet50–Vision Transformer framework and existing deep learning models over 12 training epochs.

Table 12. Classification accuracy (%) of the baseline deep learning models and the proposed ResNet50–Vision Transformer framework over 12 training epochs.

Epoch	ResNet50	VGG16	DenseNet121	Vision Transformer	Proposed ResNet50 + ViT
1	82.46	84.18	85.67	86.25	**88.18**
2	90.37	91.84	93.52	95.13	**98.25**
3	92.81	93.94	95.68	96.84	**98.78**
4	93.96	95.21	96.74	97.53	**99.12**
5	94.82	96.07	97.32	98.01	**99.18**
6	95.43	96.58	97.86	98.32	**99.27**
7	95.82	96.96	98.08	98.46	**99.35**
8	96.11	97.18	98.16	98.55	**99.41**
9	96.36	97.32	98.21	98.63	**99.51**
10	96.55	97.38	98.23	98.68	**99.55**
11	96.71	97.40	98.25	98.72	**99.65**
12	**96.82**	**97.41**	**98.26**	**98.74**	**99.68**

From the results of the experiments conducted, it can be seen that all models experienced a gradual increase in the training accuracy depending on the training epochs. It can be seen that the proposed ResNet50-Vision Transformer framework managed to attain the best accuracy in the training of the model. In the first epoch, the proposed model had a training accuracy of 88.18% compared to the other models including ResNet50 (82.46%), VGG16 (84.18%), DenseNet121 (85.67%) and Vision Transformer (86.25%). A remarkable improvement in the accuracy of the classification was seen for all models in the first training epochs. On the fourth epoch, the proposed framework had the training accuracy of 99.12% while for ResNet50, VGG16, DenseNet121, and Vision Transformer, their training accuracies were 93.96%, 95.21%, 96.74%, and 97.53%, respectively. With the rest of the epochs, the training accuracies were seen to improve gradually towards convergence. At the end of the twelfth epoch, the proposed framework managed to get the best training accuracy of 99.68% followed by Vision Transformer (98.74%), DenseNet121 (98.26%), VGG16 (97.41%) and ResNet50 (96.82%). Compared to the baseline models, the proposed framework managed to get improvements of 2.86%, 2.27%, 1.42%, and 0.94%, respectively.

5.8 Comparison of Training Loss

Table 13 shows the comparative training loss for the proposed ResNet50–Vision Transformer In order to assess the optimization performance of the proposed architecture, the training loss was compared against four popular deep learning models, which include ResNet50, VGG16, DenseNet121, and Vision Transformer, across 12 training epochs. Training loss is an indicator of the optimization performance of deep learning models during training. The following table shows the training loss achieved by the proposed architecture and the baseline architectures across the training period.

Table 13. Comparison of training loss between the proposed ResNet50–Vision Transformer and other deep learning models over 12 training epochs.

Epoch	ResNet50	VGG16	DenseNet121	Vision Transformer	Proposed ResNet50 + ViT
1	0.6932	0.6458	0.6127	0.5814	**0.5189**
2	0.2985	0.2416	0.1762	0.1024	**0.0597**
3	0.2014	0.1588	0.1193	0.0748	**0.0403**
4	0.1625	0.1213	0.0897	0.0592	**0.0312**
5	0.1382	0.1036	0.0728	0.0496	**0.0262**
6	0.1226	0.0914	0.0615	0.0435	**0.0228**
7	0.1105	0.0826	0.0537	0.0396	**0.0195**
8	0.1028	0.0763	0.0486	0.0368	**0.0164**
9	0.0961	0.0724	0.0455	0.0345	**0.0156**
10	0.0914	0.0692	0.0438	0.0327	**0.0141**
11	0.0878	0.0664	0.0415	0.0311	**0.0114**
12	**0.0842**	**0.0641**	**0.0402**	**0.0295**	**0.0105**

The experimental results reveal that the training loss for all considered models decreased steadily with an increasing number of training epochs. Out of all the examined models, the proposed ResNet50–Vision Transformer approach demonstrated the lowest training loss at all stages of the training process. In the

first epoch, the proposed approach yielded a training loss of 0.5189, while ResNet50, VGG16, DenseNet121, and Vision Transformer demonstrated training losses of 0.6932, 0.6458, 0.6127, and 0.5814, respectively. A significant decrease in training loss for all models occurred during the initial training epochs. In the fourth epoch, the training loss for the proposed approach dropped to 0.0312, which is lower than that of ResNet50 (0.1625), VGG16 (0.1213), DenseNet121 (0.0897), and Vision Transformer (0.0592). In the subsequent epochs, the training loss values steadily converged for all considered models. After the twelfth epoch, the lowest training loss was achieved by the proposed approach (0.0105), followed by Vision Transformer (0.0295), DenseNet121 (0.0402), VGG16 (0.0641), and ResNet50 (0.0842). The steady decrease in training loss across multiple training epochs shows that the proposed approach optimizes steadily.

5.9 Yolo8 Diseases Spotted Grad CAM Results

The qualitative localization ability and interpretability of the proposed system were evaluated using YOLOv8 to detect the disease-infected regions and using Grad-CAM to identify the regions of the image that have contributed to the final classification. YOLOv8 offers localization in the form of bounding boxes for the disease-infected leaf regions, whereas Grad-CAM marks the discriminative regions recognized by the classification network. Figure 8 shows some typical examples of localization and Grad-CAM for the four types of moringa leaves: (a) Healthy Leaf, (b) Bacterial Leaf Spot, (c) Cercospora Leaf Spot, and (d) Young Yellow Leaf.

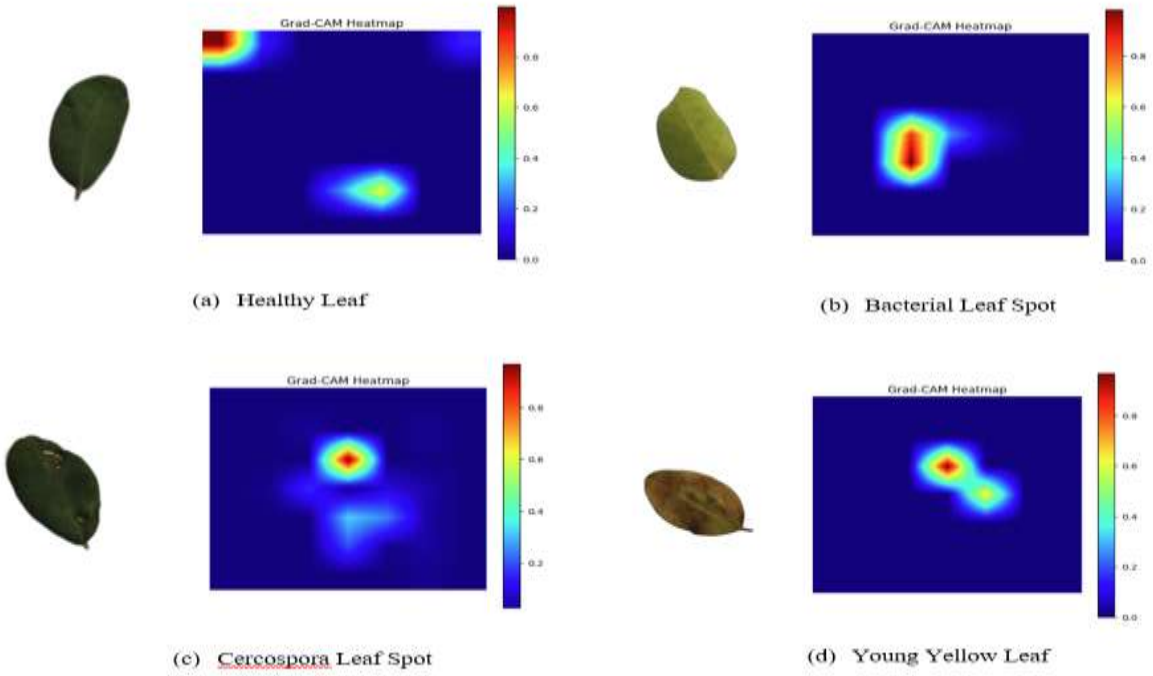


Figure 8. YOLOv8 disease localization and Grad-CAM visualization of the proposed ResNet50–Vision Transformer framework for representative moringa leaf disease classes.

The Grad-CAM heatmaps demonstrate that the proposed framework consistently concentrates on disease-relevant regions during classification. In the heatmaps, red and yellow indicate regions with high contribution to the prediction, whereas blue represents regions with minimal influence. For the Healthy Leaf sample (Figure 8(a)), only a small localized activation is observed, while most regions exhibit low activation intensity, indicating the absence of disease-related features. In the Bacterial Leaf Spot sample (Figure 8(b)), the highest activation is concentrated on the infected lesion identified by YOLOv8, with minimal activation in the surrounding healthy tissue. Similarly, the Cercospora Leaf Spot image (Figure 8(c)) exhibits strong activation over the lesion area together with lower responses in adjacent regions. For the Young Yellow Leaf sample (Figure 8(d)), the activation map is concentrated on the yellow discoloured regions corresponding to the disease symptoms identified during localization. The localization process by the YOLOv8 model correctly localizes the diseased areas before the classification process. However, the Grad-CAM output shows the classification network mainly uses the localized areas in making predictions. The localization and activation processes have shown consistency in all four disease types.

6. Discussion

6.1 Synergistic Learning via Hybrid Feature Representation

The excellent results of the proposed methodology indicate that the combination of convolutional feature extraction and transformer-based contextual learning helps achieve a more complete description of moringa leaf disease features compared to each methodology applied independently. Indeed, convolutional neural networks have high capabilities in extracting detailed visual characteristics like lesion texture, colour distribution, edge irregularities, and structure abnormalities. Nevertheless, due to the specific receptive field of CNNs, the relationships between spatially distant disease symptoms cannot be learned. At the same time, the attention mechanism of the Vision Transformer helps establish global interactions between the image patches and recognize disease patterns present throughout the whole leaf. Therefore, the combination of these two representations allows recognizing subtle differences between classes while maintaining internal consistency. This synergy helps explain why the proposed architecture has higher discriminative power since visually similar diseases often have similar colour patterns and irregular lesions' distributions. Thus, the proposed method has managed to overcome the limitations of both CNN-based and transformer-based approaches, providing a more detailed semantic representation. It is consistent with the findings of recent studies that hybrid CNN and transformer architectures provide better results for plant disease recognition compared to standard convolutional architectures. The current results have further extended the previous studies by combining the aspects of semantic segmentation, object localization, and XAI in one single diagnostic model, rather than using just image classification.

6.2 Influence of Data Preparation and Segmentation on Model Generalization

Not only does the efficacy of the proposed framework rely on the hybrid framework but it is also attributed to the data preprocessing pipeline designed specifically for the problem at hand. Agricultural imagery features high variance that can be related to various factors such as different illumination, complicated background, leaf orientation, partial occlusion, as well as varying severity of the diseases. Such variance often results in overfitting in deep neural networks trained on controlled data and poor performance on field data. By including the field collected images along with augmented samples into training process, the framework allows the network to see a wider range of visual features that helps to improve generalizability of the classifier. In addition, semantic segmentation is crucial to the feature learning process as the infected tissue has to be distinguished from healthy parts of the plant before being classified. By removing all background information, classifier is able to learn from biologically relevant regions of disease. Such approach helps to reduce ambiguity of features, filter out background noise, and increase consistency of features among various types of disease. Thus, Preprocessing, Segmentation, and Feature Extraction as a Hybrid Method Makes the Process Robust to Changes in Lighting Conditions, Complex Backgrounds, Occlusions, and Varied Degrees of Disease Severity.

6.3 Explainability, Disease Localization, and Practical Reliability

Explanation is especially significant in agricultural diagnostics since a wrong disease classification can lead to the use of the wrong pesticides and/or treatment delay. Thanks to the clear and visual explanation of each output, the proposed approach improves decision-making reliability and increases the credibility of the system among its end users. In addition to the classification results, one of the key benefits of the proposed framework is its capacity to generate interpretable and geographically meaningful outputs. It is often claimed that deep learning approaches function as black-box models and prevent people from trusting automated decisions. Using Grad-CAM, we solve this problem by highlighting those parts of the image that are most influential for classification. Instead of considering background textures or imaging artifacts as significant parts of the image, activation maps always focus on lesions, discolored tissues, and disease-specific regions. In addition, the visualizations produced through Grad-CAM can aid in validation by experts from the agricultural industry in the form of comparing regions detected to have the disease to actual physical symptoms of the pathology. This raises confidence among users, ensuring the adoption of the artificial intelligence technology into the decision-making process in agriculture. In a similar manner, the addition of YOLO localization to the framework goes a step further than traditional image classification techniques to determine the exact location of the infection on the leaf. This is information that will add value to the experts and farmers as well, as it helps in confirming the extent of the disease and allows for appropriate treatment approaches.

6.4 Scientific Contribution, Limitations, and Future Perspectives

The current study expands on previous work through the use of image pre-processing, semantic segmentation, hybrid feature learning, object detection, and explainable artificial intelligence into an all-encompassing disease diagnosis process. Most previous studies have mainly focused on image classification and deep learning models in isolation. The proposed model highlights the benefits of incorporating different components into one process. Integration allows disease diagnosis, localization, and explainability at once to increase the application of deep learning models from classification. However, there are still improvements that can be made in the current framework. The current framework has been

tested for only a few types of diseases, and more testing in geographically diverse environments of crop cultivation will provide evidence of its generalization ability. Moreover, while the hybrid approach is effective for predictions, feature learning using the transformer is computationally intensive than convolutional neural networks.

Consequently, future work will involve research efforts on model compression, knowledge distillation, lightweight attention mechanism, and edge computing deployment for implementing the model on mobile device or edge-enabled agricultural environment. Besides, the use of hyperspectral imaging and other multisensor fusion along with the temporal progression analysis of diseases may help in diagnosing diseases early, and thus will further contribute to intelligent precision agriculture. While the combination of convolutional and transformer layers is computationally complex than lightweight CNN models, however, the extra computational complexity is justified by the improved feature extraction, classification ability, and reliable diagnostics. The analysis of computational complexity has proved that the proposed approach is computationally efficient and balanced between prediction accuracy and computation time and thus can be implemented on GPU-enabled agricultural monitoring systems.

In terms of agriculture, correct disease detection early in the game helps in making interventions in a timely manner, avoids overuse of pesticides, prevents crop damage and encourages sustainable farming. Thus, the suggested framework is capable of making a contribution to intelligent crop management and precision agriculture. In general, the suggested framework is capable of proving that the integration of deep residual learning, contextual modeling with transformers, semantic segmentation, localization of diseases and explainable artificial intelligence can be considered an efficient approach to automated moringa leaf disease detection. The results of the experiments have shown that combination of complementary approaches significantly improves feature representation and prediction performance of the model.

7. Conclusion

The current work introduced a novel hybrid approach for automated disease classification of moringa leaves using U-Net based semantic segmentation, ResNet50, Vision Transformer (ViT), YOLOv8 based disease localization and Grad-CAM based Explainable Artificial Intelligence (XAI). In contrast to the traditional methods which only use either CNNs or transformers, the current approach incorporates both local features and global context into learning disease representations and classification. While semantic segmentation helps in precise detection of the infected regions of the leaves before the classification process, the YOLOv8 method contributes to disease localization, and Grad-CAM increases the transparency of the prediction results. Also, the employment of an integrated database including field images and augmented images increases the robustness of the current framework. Based on the experimental analysis, it can be stated that the proposed approach outperforms ResNet50, VGG16, DenseNet121 and the Vision Transformer itself in terms of different performance measures. It is evident that combining complementary deep learning architectures helps improve feature representation, facilitates disease class discrimination, and guarantees stable learning behavior during training. Finally, it is shown that the proposed Grad-CAM visualization proves that the model makes decisions based on biologically significant disease regions, while the YOLOv8 localization results prove that the model can reliably recognize the regions of infection on leaf images. Thus, all the obtained results confirm that the proposed model is a reliable and efficient tool for intelligent moringa leaf disease diagnosis. From the practical point of view, the proposed approach has great potential to be applied in precision agriculture since the early disease detection can help prevent crop loss and use pesticides effectively, which will increase the overall productivity.

8. Future Work

The proposed hybrid deep learning architecture offers an effective basis for intelligent crop disease diagnosis and can easily be extended to work for other crops through retraining of the model with crop-specific data. In future research, efforts will be directed towards extending the architecture to include a larger number of plant diseases and thus develop a universal system for diagnosis of plant diseases for precision agriculture. Efforts will also be made to develop lightweight hybrid models using techniques like compression, pruning, quantization, and knowledge distillation for efficient execution on mobile phones, embedded systems, and edge-AI systems while maintaining predictive accuracy. Efforts will be made to integrate hyperspectral and multispectral imaging, multimodal sensors data, IoT-based crop monitoring, and temporal analysis of disease progress. Future research directions will also be directed towards cloud-edge collaboration computing, autonomous agriculture robots, UAVs, and mobile-based decision support systems for real-time monitoring of crops health on a larger scale. These innovations will help in developing intelligent, scalable, and sustainable precision agriculture systems for early diagnosis of crop diseases and their optimized management.

References

[1]. S. A. Preanto, T. Paul, A. Khan, and M. H. I. Bijoy, "MoringaLeafNet: A multi-class leaf disease dataset for precision agriculture and deep learning research," Data in Brief, vol. 63, p. 112174, 2025. DOI: https://doi.org/10.1016/j.dib.2025.112174

[2]. A. K. Singh, B. Pandey, and S. Kumar, "Plant disease detection using deep learning: A review," Computers and Electronics in Agriculture, vol. 209, p. 107834, 2024. DOI: https://doi.org/10.1016/j.compag.2024.107834

[3]. J. Amara, B. Bouaziz, and A. Algergawy, "A deep learning-based approach for banana leaf diseases classification," in Datenbanksysteme für Business, Technologie und Web (BTW 2017) – Workshopband, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn, 2017, pp. 79–88.

[4]. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018. DOI: https://doi.org/10.1016/j.compag.2018.01.009

[5]. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, p. 1419, 2016. DOI: https://doi.org/10.3389/fpls.2016.01419

[6]. A. Upadhyay, R. Singh, and P. Kumar, "Deep learning and computer vision in plant disease detection: A review," Artificial Intelligence Review, 2025. DOI: https://doi.org/10.1007/s10462-024-11100-x

[7]. Li T, Su J and Li S (2026) Real-time detection of plant leaf diseases based on improved YOLOv13-LM in complex field environments. Front. Plant Sci. 17:1819646. doi: 10.3389/fpls.2026.1819646

[8]. R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in Proc. ICCV, 2017, pp. 618–626. DOI: https://doi.org/10.1109/ICCV.2017.74

[9]. S. R. Pydipati, T. F. Burks, and W. S. Lee, "Identification of citrus disease using color texture features and discriminant analysis," Computers and Electronics in Agriculture, vol. 52, no. 1–2, pp. 49–59, 2006.

[10]. H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik, and Z. ALRahamneh, "Fast and Accurate Detection and Classification of Plant Diseases," International Journal of Computer Applications, vol. 17, no. 1, pp. 31–38, 2011.

[11]. S. Arivazhagan, R. N. Shebiah, S. V. Nidhyanandhan, and L. Ganesan, "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features," Agricultural Engineering International: CIGR Journal, vol. 15, no. 1, pp. 211–217, 2013

[12]. S. Phadikar and J. Sil, "Rice disease identification using pattern recognition techniques," in Proc. 11th International Conference on Computer and Information Technology (ICCIT), Khulna, Bangladesh, 2008, pp. 420–423.

[13]. A. Camargo and J. S. Smith, "Image pattern classification for the identification of disease causing agents in plants," Computers and Electronics in Agriculture, vol. 66, no. 2, pp. 121–125, 2009.

[14]. P. Revathi and M. Hemalatha, "Advance computing enrichment evaluation of cotton leaf spot disease detection using image edge detection," in Proc. International Conference on Emerging Trends in Science, Engineering and Technology, 2012, pp. 1–5.

[15]. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," Frontiers in Plant Science, vol. 7, Art. no. 1419, 2016.

[16]. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018.

[17]. E. C. Too, L. Yujian, S. Njuki, and Y. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," Computers and Electronics in Agriculture, vol. 161, pp. 272–279, 2019.

[18]. A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70–90, 2018.

[19]. J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," Biosystems Engineering, vol. 172, pp. 84–91, 2018.

[20]. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in Proc. 3rd Int. Conf. Learning Representations (ICLR), San Diego, CA, USA, 2015.

[21]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778.

[22]. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 4700–4708.
[23]. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint, arXiv:1704.04861, 2017.
[24]. M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proc. 36th Int. Conf. Machine Learning (ICML), Long Beach, CA, USA, 2019, pp. 6105–6114.
[25]. J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018, pp. 7132–7141.
[26]. S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in Proc. European Conf. Computer Vision (ECCV), Munich, Germany, 2018, pp. 3–19.
[27]. E. C. Too, L. Yujian, S. Njuki, and Y. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," Computers and Electronics in Agriculture, vol. 161, pp. 272–279, 2019.
[28]. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018
[29]. J. G. A. Barbedo, "Plant disease identification from individual lesions and spots using deep learning," Biosystems Engineering, vol. 180, pp. 96–107, 2019.
[30]. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale," in Proc. Int. Conf. Learning Representations (ICLR), 2021.
[31]. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), 2021, pp. 10012–10022.
[32]. H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing Convolutions to Vision Transformers," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), 2021, pp. 22–31.
[33]. B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou, and M. Douze, "LeViT: A Vision Transformer in ConvNet's Clothing for Faster Inference," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), 2021, pp. 12259–12269.
[34]. J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation," arXiv preprint, arXiv:2102.04306, 2021.
[35]. A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. Roth, and D. Xu, "UNETR: Transformers for 3D Medical Image Segmentation," in Proc. IEEE Winter Conf. Applications of Computer Vision (WACV), 2022, pp. 574–584.
[36]. H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training Data-Efficient Image Transformers & Distillation Through Attention," in Proc. Int. Conf. Machine Learning (ICML), 2021, pp. 10347–10357.
[37]. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in Proc. European Conf. Computer Vision (ECCV), 2020, pp. 213–229
[38]. M. Murugavalli, V. S. Kumar, R. Karthikeyan, and P. Ramesh, "Hybrid CNN–Vision Transformer Framework for Plant Disease Classification," Computers and Electronics in Agriculture, vol. 228, 2025.
[39]. A. Bouacida, S. Rahman, and M. Ibrahim, "Vision Transformer-Based Intelligent Crop Disease Detection under Field Conditions," Artificial Intelligence in Agriculture, vol. 11, pp. 85–99, 2025.
[40]. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 2015, pp. 234–241.
[41]. Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A Nested U-Net Architecture for Medical Image Segmentation," in Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA), 2018, pp. 3–11.
[42]. O. Oktay et al., "Attention U-Net: Learning Where to Look for the Pancreas," arXiv preprint, arXiv:1804.03999, 2018.
[43]. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.
[44]. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint, arXiv:1804.02767, 2018.

[45]. G. Jocher et al., “YOLOv5,” GitHub Repository, Ultralytics, 2021.
[46]. G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLOv8,” GitHub Repository, Ultralytics, 2023.
[47]. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 618–626.
[48]. M. T. Ribeiro, S. Singh, and C. Guestrin, “Why Should I Trust You? Explaining the Predictions of Any Classifier,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2016, pp. 1135–1144.
[49]. S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in Proc. 31st Conf. Neural Information Processing Systems (NeurIPS), 2017, pp. 4765–4774.
[50]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[51]. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Proc. Int. Conf. Learning Representations (ICLR), 2015.
[52]. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4700–4708.
[53]. A. Dosovitskiy et al., "An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale," Proc. Int. Conf. Learning Representations (ICLR), 2021.
[54]. W. K. Pratt, Digital Image Processing: PIKS Scientific Inside, 4th ed. Hoboken, NJ, USA: Wiley, 2007.
[55]. R. G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981.
[56]. mondal, kartick ; Sarkar, Tanmay (2024), “Moringa leaves”, Mendeley Data, V1, doi: 10.17632/96xjz92ttj.1