Journal of Theoretical and Applied Information Technology
30th April 2026. Vol.104. No.8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# TIGLEAFNET-Q: AN EXPLAINABLE HYBRID QUANTUM CLASSICAL FRAMEWORK FOR MULTI-CROP LEAF DISEASE DETECTION WITH ENHANCED FEATURE REPRESENTATION

**D. VETRITHANGAM[1], KARTHIKEYAN T[2], VIVEKANANDAN S J[3], ASHWINI BARBADEKAR[4]**

**VASUKIDEVI G[5], MUTHUKUMAR SUBRAMANIAN[6], ASWATHY S NAIR[7]**

[1]Department of Computer Science & Engineering, University Institute of Engineering, Chandigarh University, Mohali, 140413, India.
[2]Department of Computer Science and Business Systems, Panimalar Engineering College, Chennai, India.
[3]Department of Computer Science and Engineering, Dhanalakshmi College of Engineering, Chennai, India.
[4]Department of Electronics and Telecommunications Engineering, Vishwakarma Institute of Technology, Pune, India.
[5]Department of Computer Science Engineering, RMK Engineering College, Kavaraipettai, India.
[6]Department of CSE, Hindustan Institute of Technology and Science, Deemed University, Chennai, India.
[7]Department of Computer Applications, Marian College Kuttikkanam Autonomous, Kuttikkanam, Kerala, India.
E-mail: [1]vetrigold@gmail.com ,[2]karthi4cse@gmail.com , [3]saivivekphd@gmail.com,
[4]ashwini.barbadekar@vit.edu , [5]vasukidevi@gmail.com,[6]drsm.iiit@gmail.com,
[7]aswathy.s@mariancollege.org

**ABSTRACT**

This study addresses the critical need for accurate and interpretable plant disease detection in sustainable agriculture. Existing deep learning models often suffer from limited generalization across crops, insufficient modeling of complex feature interactions, and a lack of explainability, restricting their real-world applicability. To overcome these challenges, this work proposes TIGLeafNet-Q, a hybrid quantum–classical framework for multi-crop leaf disease detection. The proposed approach integrates an InceptionV3 transfer learning backbone for deep feature extraction, followed by feature compression and angle encoding into quantum states. A variational quantum circuit (VQC) is employed to capture higher-order correlations among disease features, and Grad-CAM is used to provide visual interpretability. The model is evaluated on large-scale tomato (22,980 images) and cotton (2,204 images) datasets. Experimental results demonstrate classification accuracies of 99.28% (tomato) and 97.23% (cotton), outperforming existing methods. The proposed framework contributes a scalable, explainable, and multi-crop capable solution, highlighting the potential of hybrid quantum–classical learning in agricultural intelligence systems.

**Keywords:** *Transfer Learning; Tomato; Cotton; Leaf Disease Detection; Deep Learning; Hybrid Quantum-Classical Learning; Angle Encoding; Variational Quantum Circuit*

## 1. INTRODUCTION

The agriculture sector is crucial for generating income and supplying the public with food. As a result of the increased need for food, agriculture is severely strained. By exporting agricultural products to other nations, a country with good agricultural land may both meet its own food needs and contribute to the economy. The need for food increases along with the global population [1]. Vegetables play a remarkable role in both the agriculture and horticulture sectors of India. Vegetable production in India has dramatically expanded during the past few years. Vegetables are fast-growing, high-yielding crops that are both economically viable and nutrient-dense. In terms of horticultural product production, India ranks second worldwide. This is due to India's diverse geography and a great variety of climates. A vital cash crop that aids in the creation of natural fiber is cotton. Making clothes is the cotton crop's main contribution. It promotes the expansion of the textile sector. Plant protection is important for cotton production, in addition to other things. A nation frequently

struggles to meet demand for food because of sickness in the agricultural industry. The detection of plant diseases is a crucial factor that has been explored throughout the years and is driven by the requirement to create nutritious food [2]. Anyhow, there are several desirable factors to consider, such as affordability, usability, sensitivity, and accuracy. The tomato, also known as Solanum lycopersicum (L.), was once known as Lycopersicum esculentum (Mill.), and it is a member of the genus Lycopersicon in the community of Solanaceae. Tomatoes are a widely grown and significant crop commercially around the world. Because of their high nutrient content, particularly lycopene, vitamin C, and derivatives of hydroxycinnamic acid, tomatoes are regarded as "protective crops." Tomatoes contain lycopene, which has antioxidant and anti-cancer properties.

The cultivation and production of tomatoes in India have been affected by the severe viral disease Tomato Leaf Curl Disease (ToLCD) [3]. ToLCD can be easily identified by prominent leaf curling, cupping, and shrinkage of the leaf's size, stunted growth with poor flower formation, and fruit setting in plants. The potential for developing antiviral techniques to combat viral infection is expanded by an understanding of basic biology, such as the replication of Gemini viruses. Plant vectors like leafhoppers and whiteflies are used by Gemini viruses to spread their closed circular DNA and the resulting virion particles from one plant to another: ToLCBDV (TLC Bangladesh Virus), ToLCNDV (TLC New Delhi Virus), and ToLCVSLV (TLC Sri Lanka Virus) [4][5]. Tomato plants have been known to contract Gemini viruses from at least 57 different species [6]. Surveys for begomovirus-linked diseases in okra fields in India between 2005 and 2009 found widespread infection with yellow vein mosaic disease and extremely diverse symptoms, making it challenging to visually identify the disease-associated begomoviruses [7][8]. Due to the absence of the begomovirus's full-length sequence, which is necessary for begomovirus nomenclature, the precise identity of the begomovirus linked to this disease has not been determined. Infected plants showed the yellow leaf curl sign in central Thailand, where the illness of ash gourds caused by the begomovirus was also documented [9]. Additionally, the begomovirus-infected tobacco plants used in the current study displayed a variety of symptoms ranging from minor to major. The increased significance of begomoviruses to agriculture in recent years appears to be the result of a number of causes, including numerous infections, recombination, and interspecific synergism. It's interesting to note that begomovirus beta satellite complexes don't seem to cause issues in cucurbits. The presence of cognate beta satellites boosted both the number of accumulating helper viruses and the accumulation of symptoms. Intriguingly, intra-species recombinants were discovered in the majority of the begomoviruses. Three viruses dominated: ToLCNDV, a bipartite virus; ToLCBaV; and monopartite viruses. Analysis of about 50 samples shows a strong correlation between the two bipartite and monopartite begomoviruses and the sickness in the study area [10][11][12]. Twelve distinct species of tomato leaf disorder viruses, several of which are seen in combined infections, cause tomato leaf disorder and yellow leaf disorder diseases [13]. PCR has been frequently employed as an assay for the purpose of finding plant viral genomes. When compared to the traditional ELISA approach, it has yielded results that are sensitive. Following the creation of degenerate universal primers, PCR is increasingly used to detect begomoviruses. These primers were created with a modification in one or more places based on the highly similar area seen in all begomoviruses. Seven hosts in Africa, India, America, and Europe have had known or potential WTGs identified using PCR with these degenerate primers. When compared to hybridization approaches, this technique's sensitivity yielded results that were many times more accurate. When compared to PCR- or antibody-based detection methods, RCA has been employed as a less expensive method. The RCA method was regarded as a trustworthy method for detecting Gemini viruses in natural sources of plants. In the 1980s, RCA was employed to find WTGs in a single plant in West Africa. RCA-amplified DNA products were characterized, and sequencing approximately 900 bases at a time, cloning, and plasmid purification were also performed [14].

The detection procedure is carried out by Faster R-CNN in two steps. A picture is entered into a Region Proposal Network (RPN) in the first step, where a feature extractor processes the image. Each suggested object has a score, and features with a medium level of difficulty are used to forecast it. When training RPNs, the system assesses whether an anchor includes an object or not, depending on the connection units through the objects and the actual data. The next stage involves cropping features from the same feature map using the previously generated box proposals. The subsequent layers of the feature extractor are then fed these cropped features in order to forecast the class and bounding box for each area proposal. As the overall procedure takes place on a

single unified network, the system is able to endure full-image convolutional characteristics with the detection network, making it possible to almost instantly generate regional proposals without spending any money [15]. Visual data is commonly trained or analyzed using CNN, a neural network technology. The matrix structure of convolution is used to filter the images. The CNN uses the input layer, fully connected layer, convolutional layer, and pooling layer to construct the CNN, and eventually the connected dataset division layer for data training. It can provide each layer of the input test set with a set of computations to perform. ToLCGV is among the most harmful begomoviruses connected to tomato leaf disorder disease in India. Given the global climate change and its impact on leaf curl infections, it is possible that it will spread to hosts other than tomatoes [16]. Consequently, the analysis of virus attachment, entry, interpretation, transcription, and effects on host cells is presently focusing on developing antiviral and/or disease-improving techniques [17]. The most significant plant species affected by Cotton Leaf Curl Disease (CLCuD) are those in the family Malvaceae (Genus: Gossypium) [18]. In Maharashtra's tomato-growing regions in 2008, a significant outbreak of the leaf curl disease resulted in crop losses of 69–100%. With signs like severe leaf curving, leaf deflection, dryness, and plant mortality, leaf curl-affected plants lost between 68% and 89% of their yield. The chili plant is suffering greatly from curly leaf disease. As per the report of the Food and Agriculture Organization, there will be a rise in the world's population of around 2.3 billion people. Residual Network (ResNet) and Inception transfer learning. The model calculation has been streamlined by ResNet. Plants face several challenges, even though they are essential to life. Product quality and quantity suffer from a lack of systematic illness identification. This has more negative effects on a nation's economy. According to earlier studies, 80–90% of plant diseases manifest on leaves. An effective solution is needed for leaf disease problems, as identifying affected plants and diagnosing various disease types on farms can be time-consuming. Farmers may not accurately assess the type of plant disease [19]. This choice can result in the plant implementing an insufficient and ineffective defense. The wrong treatment of the disease can result in a 50% reduction in plant yield. Contrarily, disease-prevention substances like fungicides and bactericides have a detrimental effect on the agricultural ecosystem. A hybrid leaf disease classification framework combining advanced preprocessing, deep feature extraction, feature optimization, quantum-enhanced classification, and explainable AI was proposed for robust tomato and cotton disease detection[49]. Quantum dots (QDs) are nanoscale semiconductors with unique optical properties used in advanced imaging and sensing. They aid plant disease detection and management through efficient pathogen tracking and diagnosis [52].

Despite recent progress, existing leaf disease detection methods still suffer from several important technical limitations. Many models are constrained to single-crop or limited disease classification settings, reducing their applicability in real agricultural environments. Furthermore, their performance degrades under noisy and variable imaging conditions, such as illumination changes, blur, and background interference. In addition, purely classical deep learning techniques often fail to effectively model complex higher-order correlations among disease-specific features. Another critical limitation is the lack of interpretability, which restricts trust and adoption in practical agricultural decision-making systems.

**Problem Statement**

Despite the advancements in deep learning-based plant disease detection, there remains a lack of scalable and interpretable models capable of generalizing across multiple crops while effectively capturing complex feature interactions. Existing approaches are limited by their dependence on single-crop datasets, insufficient modeling of higher-order correlations, and a lack of explainability, which collectively hinder their deployment in real-world agricultural settings.

**Research Questions**

To address these challenges, this study investigates the following research questions:

RQ1: Can hybrid quantum–classical learning enhance feature representation for plant disease classification?

RQ2: How effectively can a unified framework generalize across multiple crops, such as tomato and cotton?

RQ3: Can explainability be preserved within a hybrid quantum-enhanced architecture?

Motivated by these challenges, this study proposes TIGLeafNet-Q, a hybrid quantum–classical explainable framework that integrates InceptionV3-based transfer learning, feature compression, angle encoding, variational quantum

circuits, and Grad-CAM to achieve robust, accurate, and interpretable multi-crop leaf disease detection.

The major contributions of this work are as follows:

• A hybrid quantum–classical architecture is developed for multi-crop leaf disease detection.

• InceptionV3 is employed as a transfer-learning backbone for robust deep feature extraction.

• A feature compression and angle-encoding strategy is introduced to map classical CNN features into quantum states.

• A variational quantum circuit is incorporated to model higher-order feature interactions before classification.

• Grad-CAM is used to preserve explainability by highlighting disease-relevant image regions.

**Organization of Paper**

The rest of the paper is organized as follows: Section 2 provides a thorough analysis of current leaf disease detection and classification methods. Section 3 covers the materials and methods used in the proposed model. Section 4 explains the system model, architecture, and working principles of the proposed model. Section 5 explains the results produced by the proposed model and provides a detailed comparative analysis. Section 6 concludes the paper with future scope.

## 2. LITERATURE REVIEW

The tomato is among the most widely consumed vegetables in the world and is arguably the best garden crop. One of the strongest antioxidants found in dietary carotenoids, lycopene pigment, which is found in tomatoes, is a carotenoid. Tomato consumption has been connected to a lower risk of developing chronic conditions like cancer and cardiovascular disease. Plants play a crucial role in our existence because they give us vegetables and protect us from harmful radiation. Without plants, life could not exist since they sustain all terrestrial life and safeguard the ozone layer, which protects against UV rays. A popularly grown, nutrient-rich vegetable is the tomato plant. The significance of plants to human existence is very important as they give us food and shield us from hazardous radiation. Without plants, life could not exist since they sustain all planetary life and safeguard the ozone layer, which filters UV rays. A popularly grown, nutrient-rich vegetable is the tomato plant. Plant diseases provide a vast area for research with a focus on the biological traits of diseases. These days, diagnosing plant diseases can be challenging and requires extra caution. Early detection of these diseases maintains the benefits of tomatoes that consumers expect. Recently, hybrid quantum–classical machine learning approaches have emerged as a promising direction for enhancing feature representation and optimization in image classification tasks. In such frameworks, classical deep neural networks are typically used for feature extraction, while quantum encoding and variational quantum circuits are employed for feature transformation or classification. These methods aim to exploit the representational capacity of quantum state spaces to capture correlations that may be difficult for purely classical models to learn efficiently. However, the application of hybrid quantum–classical learning to plant leaf disease detection remains limited, particularly in multi-crop and explainable disease classification settings. The manual implementation of these steps is a conventional approach for disease management and disease assessment in tomatoes. By creating a computerized process to complete all of these chores for the user, aggravation and time spent performing them generally will be eliminated [20]. Cohen et al. [21] developed a system for the effective detection of the Gemini virus spread by whiteflies. The fast and precise identification of the causing virus will allow for the proper management of TOLCD epidemics. Nucleic acid and serological methods have been employed as accurate and sensitive diagnostic techniques for the identification of WTGs in ToLCD. Kushwaha et al. [22] recommended that, for the direct identification of the Gemini virus in the infected plants, a number of molecular detection techniques have been modified. These approaches include nucleic acid hybridization and polymerase chain reaction-based ones. Kushwaha et al. [23] designed CNNs, which are among the machine learning field's most promising categorization methods. The most alluring aspect of CNN is its capacity to automatically extract the necessary elements for the categorization of the images throughout its learning operations. In recent months, CNN has displayed outstanding performance in complex general-image classification tasks. After analyzing images of tomato leaves, we proposed an innovative method for detecting diseases in tomato plants. This approach empowers farmers to identify plant diseases without relying on agricultural experts. As a result, they can take timely action to treat affected crops, ultimately enhancing both the yield and quality of food production, while also increasing their agricultural profitability.Turkoglu et al. [24]

developed a CNN algorithm that correctly diagnoses 100 % of spider mites, 100 % of leaf minor, 94 % of healthy plants, and 98 % of bacterial light. Additionally, the researcher has removed noise using a variety of preprocessing methods. The categorization of the outcome into bacterial blight, healthy, and leaf miner is the main cause. This paper identified problems with existing models that performed well on a single type of leaf disease dataset, but the same model produced different results if used for detecting other plant diseases. The accuracy and other performance metrics produced by the existing models or algorithms are lower. Azath et al. [25] proposed a model for the identification and classification of leaf diseases, demonstrating strong effectiveness through both qualitative and quantitative analyses. The model achieved a high accuracy of 99.98% when evaluated on a single dataset, indicating its reliability for leaf disorder detection, although its performance may be limited by the use of a single dataset.Albattah et al. [26] used a deep-CNN model to identify apple leaf disease and produced 97.14 % accuracy. This model worked well only on the apple leaves. Pantazi et al. [27] used CNN and VGG Net architecture to classify the various classes of tomato leaves and achieved 97 % accuracy; however, this method struggles to produce good accuracy with large datasets. Agarwal et al. [28] proposed a CNN-based model with three convolutional and pooling layers followed by two fully connected layers for tomato leaf disease classification. The model outperformed pre-trained networks such as VGG16, InceptionV3, and MobileNet, achieving an average accuracy of 91.2% across nine disease classes and one healthy class. The model works well with clean, balanced training data, but its real-world performance in noisy or diverse conditions is unvalidated. So far, proving this study's real-world usefulness and reliability remains elusive. Abulizi et al. [29] presented DM-YOLO, a YOLOv9 variant designed for detecting disease in tomato leaves in realistic environments; this variant outperformed the base original and offers 92.5% precision, 95.1% average precision, and 86.4% mAP. The dynamic upsampling mechanism, which leads to this improvement, is DySample coupled with MPDIoU loss calculation. It effectively dealt with overlapping symptoms, tiny lesion regions, and differences in illumination types. Osmenaj et al. [30] have also mitigated the issue by training a custom CNN on the Kaustubh B. The Tomato Leaf Disease Detection 2020 dataset was tested with real-world images from Greece to ensure reliability for different conditions. The dataset contains 11000 un-augmented images. According to this study, removing the background improves accuracy. The hybrid architecture of CNN-VGG achieved 98% accuracy. Navya and Rao [48] proposed a hybrid model using QViT for tomato leaf disease detection. It tackles the slow convergence of classical optimization methods. Quantum-based optimization allows for faster parameter search. The model achieved 98.5 percent accuracy and cut the training time by almost two-thirds. Genemo M [50] proposed a quantum convolutional neural network (QCNN)-based approach that integrates quantum computing with deep learning for automated blight disease detection and classification. The model enables efficient and early identification of the disease and achieves a high accuracy of 96.9%, demonstrating its potential for practical agricultural applications. Gumpula Jhansi et al. [51] proposed a QCNN-based model for plant disease detection, achieving 94.96% accuracy with only 9,914 parameters on the PlantVillage dataset. The results demonstrate its efficiency and potential for accurate, low-complexity agricultural applications.

This study advances tomato disease detection by broadening dataset diversity, increasing real-world relevance, and boosting overall efficiency. Table 1 reviews existing techniques for detecting and classifying leaf diseases. It is clear from Table 1 that performance improvement is still required, both in terms of classification performance and accuracy.

Overall, recent advances in plant disease detection have predominantly relied on deep convolutional neural networks such as VGG, ResNet, and InceptionV3. Although these models achieve high accuracy under controlled experimental conditions, their performance often degrades in real-world agricultural environments due to sensitivity to noise, illumination variations, blur, and dataset bias. Furthermore, many existing approaches rely on limited or single-crop datasets, thereby restricting their applicability in practical agricultural scenarios where multiple crops are cultivated simultaneously. In addition, purely classical deep learning models may not effectively capture complex higher-order feature interactions inherent in disease patterns, which can limit their classification capability. Several studies have attempted to mitigate these challenges through transfer learning and hybrid approaches; however, these methods still exhibit limitations in generalization, robustness, and scalability. Another critical concern is the lack of interpretability, which reduces trust and hinders deployment in precision agriculture systems where explainability is essential for decision-making.

Emerging hybrid quantum–classical learning models offer promising potential for enhancing feature representation through quantum state encoding and variational circuits, enabling improved modeling of complex feature dependencies. Nevertheless, their application in plant disease detection remains largely underexplored, particularly in multi-crop and explainable settings. Therefore, there is a clear lack of frameworks that simultaneously achieve multi-crop generalization, effective higher-order feature modeling, and explainability for trustworthy decision-making. To address these limitations, this study proposes TIGLeafNet-Q, a hybrid quantum–classical explainable framework designed to provide robust, scalable, and interpretable multi-crop leaf disease detection.

## 3. Materials and Methods

### 3.1 Materials

#### 3.1.1 Dataset

The research dataset contains 10 distinct groups of tomato leaf images, including nine disease types and one healthy leaf category. Each image receives labels and then gets distributed between training, validation, and test sections, which enables the evaluation of disease classification machine learning models. The dataset contains the following categories and their image counts Bacterial Spot 1,702 training 425 validation 5 test images Early Blight 1,920 training 480 validation 5 test images Late Blight 1,851 training 463 validation 5 test images Leaf Mold 1,882 training 470 validation 5 test images Septoria Leaf Spot 1,745 training 436 validation 5 test images Spider Mites Two-Spotted Spider Mite 1,741 training 435 validation 5 test images Target Spot 1,827 training 457 validation 5 test images Tomato Mosaic Virus 1,790 training 448 validation 5 test images Tomato Yellow Leaf Curl Virus 1,961 training 490 validation 5 test images and Tomato Healthy 1,926 training 481 validation 5 test images. Table 2 shows the dataset's composition: 18,345 training images, 4,585 validation images, and 50 test images, totaling 22,980 images. We can build and test accurate models for tomato leaf disease using this dataset. Tomato leaf disease images are categorized in Figure 1. You can find the tomato leaf disease dataset at https://www.kaggle.com/datasets/shylesh101/tomato-leaf-disease.

*Table 2. A dataset with different numbers of classes and their numbers*

| Category | Train dataset (No. of images) | Validation dataset (No.of images) | Test dataset (No. of images) |
|---|---|---|---|
| Bacterial__spot | 1702 | 425 | 5 |
| Early_blight | 1920 | 480 | 5 |
| Lateblight | 1851 | 463 | 5 |
| Leaf_Mold | 1882 | 470 | 5 |
| Septoria_leaf_spot | 1745 | 436 | 5 |
| Spider__mites Two-spottedspider_mite | 1741 | 435 | 5 |
| Target Spot | 1827 | 457 | 5 |
| Tomatomosaic_virus | 1790 | 448 | 5 |
| Tomato yellow leaf curl virus | 1961 | 490 | 5 |
| Tomato__healthy | 1926 | 481 | 5 |
| Total | 18345 | 4585 | 50 |

#### 3.1.2 Data Pre-Processing

The original preprocessing pipeline of TIGLeafNet is preserved in the proposed model. In this context, the sizes of the images of tomatoes and cotton for the deep neural network were made 224 × 224 × 3 pixels to meet the input dimension specifications of InceptionV3. Even if the intermediate conversion is in grayscale, the final model input is in RGB. The contrast is enhanced, and subtle lesion patterns could be observed in different illumination environments by using CLAHE. Gaussian blur is used to reduce noise and smooth minor pixel variations to avoid losing other structure details. Dividing the pixel values by 255 ensures that the values lie between 0 and 1, thus standardizing the data and speeding up training. To facilitate auxiliary studies, grayscale images can be converted to binary images to segregate the infected regions from the background. Furthermore, the adoption of augmentation methods like rotation and flipping was done to enhance the dataset diversity and improve generalization. This is a pipeline that preprocesses the input image to extract the features. Figure 2 shows a preprocessing of an early blight tomato leaf image. The most basic step is where the required features are retained. The steps applied here are grayscale, which retains the most important features. The converted image is passed through CLAHE to

enhance the contrast between various features, resulting in better visibility of features. After that, we apply a Gaussian blur to the image to remove noise. Normalization makes pixel values consistent and eases model training. Thresholding is the process of creating a black-and-white image to emphasize the most important areas in an image. In the end, the picture is turned and flipped in order to create more variations, which are better for learning. After applying all the preprocessing techniques to the original input images, the following images were generated (as shown in Figure 3). The images represent the respective tomato leaves with the diseases of Bacterial Spot, Early Blight, Late Blight, Leaf Mold, Septoria Leaf Spot, Spider Mites, Target Spot, Tomato Yellow Leaf Curl Virus, and Tomato Mosaic Virus. These binary (black and white) images help to highlight disease patterns more clearly and are useful for training machine learning models for automatic disease detection.

### 3.2 Methods

#### 3.2.1 Transfer learning using InceptionV3

Transfer learning is the process of integrating previously learned information from prior experiences to solve related new issues. By leveraging their past weights to train on new problems, this technology allows learned deep learning architectures to be reused. In neural networks, only the hidden layers starting from the output layer in the reverse direction are permitted to learn from the data. This permits these layers to partially freeze and prevent them from altering their weights and biases during training. In situations where existing models have been trained on a large amount of data, transfer learning is beneficial. By only training a portion of the model rather than the entire model, computation time can be reduced. (Yang, n.d.). The mathematical model of transfer learning can be described as follows: Suppose we have two tasks, Task A and Task B. Let X be the input space, Y_A be the output space for Task A, and Y_B be the output space for Task B. Task A uses f_A: X→Y_A, Task B uses f_B: X→Y_B, both handling input-output mapping. Transfer learning assumes a relationship exists between Task A and Task B, where knowledge from Task A can benefit Task B. This relationship can be represented as a function g: Y_A → Y_B mapping the output space of Task A to the output space of Task B. Transfer learning, mathematically, trains a model, f_A, on Task A given labeled dataset D_A, comprised of pairs (x_1, y_A_1) to (x_n, y_A_n). Using what we learned from Task A, we change function f_A to f_B, improving how well the model does on Task B. This can be achieved by training f_B on a smaller labeled dataset D_B = {(x_1, y_B_1)... (x_m, y_B_m)}, where y_B_i = g(y_A_i) for all i. Function g maps Task A labels to Task B labels. The adapted model f_B can then be used to make predictions on new inputs for Task B. Transfer learning's efficacy hinges on Task A and Task B's relatedness and function g's ability to bridge them.

#### 3.2.2 InceptionV3

Inception v3, a deep convolutional neural network, processes images through a series of sequential mathematical operations. An input image has dimensions W × H × C, where W and H are the image's width and height, and C is the number of channels (e.g., 3 for RGB images). This input goes through a series of layers, each performing a specific computational function. The initial layer in Inception v3 is a convolutional layer that applies filters to the input image. Let X be the input image and K be the set of filters, where each filter K_i has a size of k_i x k_i x C The convolutional layer computes feature maps through a ReLU activation on the convolution of input X with kernel K, yielding Y. Once the convolutional layer has processed the input the resulting feature map is then fed through a series of Inception modules Each Inception module uses several convolutional operations with different filter sizes, plus pooling layers, which lets it extract rich features at many scales Each module transforms an input feature map Y to an output feature map Y'=f(Y). The final Inception module's output Y″ is then globally average-pooled. This layer calculates the average value for each channel in the feature map, producing a feature vector v of length C, where C is the number of channels. The vector v is then fed into a fully connected layer, which performs a linear transformation given by u = Wv + b, where W is the weight matrix and b is the bias vector. Lastly, a SoftMax activation function is applied to the output u to generate the final probability distribution over the target classes.

#### 3.2.3 Hybrid Quantum Feature Encoding and Transformation

To overcome the representational limitations of purely classical classifiers, the proposed TIGLeafNet-Q framework incorporates a hybrid quantum–classical learning module after deep feature extraction. The high-dimensional feature vector generated by InceptionV3 is first compressed through a dense projection layer to reduce

redundancy and align the dimensionality with the number of available qubits. The compressed feature vector is then encoded into quantum states using angle encoding, where each feature value controls the rotation angle of a parameterized quantum gate. A Variational Quantum Circuit (VQC), consisting of trainable rotation and entanglement layers, is subsequently applied to transform the encoded features and capture higher-order disease-specific correlations. The quantum measurement outputs are then passed to classical dense layers for final disease classification.

## 4. PROPOSED METHODOLOGY

### 4.1 System model

InceptionV3 is an improved version of the original Inception architecture, first introduced as GoogLeNet in 2014, offering greater depth, improved accuracy, and better computational efficiency compared to InceptionV1 and V2. InceptionV3 consists of 48 convolutional layers organized into Inception modules, interleaved with max-pooling and average-pooling layers, followed by fully connected layers for classification. When the model is instantiated in the proposed system, pre-trained weights from ImageNet are automatically loaded by setting weights='imagenet'. In this work, InceptionV3 is used as a transfer-learning backbone rather than trained from scratch. The initial convolutional and Inception blocks (approximately the first 30 convolutional layers) are frozen to preserve generic low-level and mid-level visual features learned from ImageNet, while the deeper Inception blocks (the remaining 18 convolutional layers) are unfrozen and fine-tuned on tomato and cotton leaf images to capture disease-specific patterns. The original top classification layer of InceptionV3 is removed by setting include_top=False. The feature maps produced by the convolutional backbone are progressively downsampled through pooling layers, reducing spatial dimensions while preserving semantically rich features. The final feature map produced is 7 × 7 × 2048, which is then transformed into a one-dimensional deep feature vector. In the original TIGLeafNet model, this feature vector is directly passed to Dense layers for multi-class classification. However, unlike the original TIGLeafNet architecture, the proposed TIGLeafNet-Q framework extends the classical transfer-learning pipeline by integrating a hybrid quantum feature transformation stage. After feature extraction using InceptionV3, the resulting deep feature vector is first compressed to a lower-dimensional representation and then encoded into quantum states using angle encoding. Initially, the encoded features are applied to a variational quantum circuit, which is then classified classically. This design architecture allows our model to leverage the strong visual representation ability of Inceptionv3 while maintaining improved modeling interaction between features using quantum circuits.Thus, the proposed TIGLeafNet-Q system consists of three major modules: (1) dataset preparation and preprocessing, (2) InceptionV3-based deep feature extraction with selective layer unfreezing, and (3) hybrid quantum–classical classification using feature compression, angle encoding, a variational quantum circuit, and a classical dense output layer for tomato and cotton leaf disease detection.

### 4.2 Architecture and working

The proposed TIGLeafNet-Q architecture was designed for automated and explainable leaf disease detection on 22,980 tomato leaf images (20,568 diseased across 10 classes and 2,412 healthy) and 2,204 cotton leaf images (1,951 training, 200 validation, and 253 test samples across three disease classes and one healthy class). All input images were standardized to 224 × 224 × 3 pixels to satisfy the input requirements of InceptionV3. To improve interpretability, Grad-CAM was employed to generate saliency maps highlighting the regions of the leaf images that most strongly influenced the classification decisions. Unlike the original TIGLeafNet architecture, which relied solely on a classical Dense classification head, the proposed TIGLeafNet-Q framework introduces a hybrid quantum–classical learning strategy. The architecture consists of five sequential stages: (1) image preprocessing, (2) deep feature extraction using InceptionV3, (3) feature compression, (4) quantum feature encoding and transformation, and (5) hybrid classification through classical dense layers. This design enables the model to combine the strong visual feature extraction capability of InceptionV3 with the enhanced feature interaction modeling of variational quantum circuits. In the first stage, all images are resized to 224 × 224 pixels while maintaining the three-channel RGB format, such that the final input to the network remains 224 × 224 × 3. To improve image quality and visibility of diseases, contrast-limited adaptive histogram equalization (CLAHE) is applied to produce a significant enhancement in local contrast, increasing the visibility of subtle lesions or patterns under varying illumination. Subsequently, a Gaussian blur is applied to eliminate any high-frequency noise while maintaining important structural details like

spots, edges, and textures. Equation (1) describes image normalization using channel-wise standardization with ImageNet statistics.

$$x' = \frac{x - \mu_{channel}}{\sigma_{chann}} \quad (1)$$

where $\mu_{channel}$ and $\sigma_{channel}$ denote the mean and standard deviation for each RGB channel. Binary thresholding, For example, Otsu's method, can be applied on intermediate grayscale representations to emphasize diseased region area but not the final input to InceptionV3. Increasing the number of images with random rotations, flips, zoom, and brightness will help the model be less sensitive to changing angles and lighting. They're crucially, all augmented images have a 224 × 224 × 3 input size. In the second stage, the preprocessed images are passed through the InceptionV3 backbone, which serves as the transfer-learning feature extractor. The architecture consists of 48 convolutional layers, of which the first 30 layers are frozen to preserve generic low-level and mid-level visual features learned from ImageNet, while the remaining 18 layers are fine-tuned on tomato and cotton leaf images to capture disease-specific characteristics. Each Inception block applies convolutional filters of multiple sizes, such as 1 × 1, 3 × 3, and 5 × 5, together with pooling operations in parallel, enabling efficient extraction of multi-scale disease features. The convolution operation can be mathematically expressed in equation(2).

$$Y_{i,j}(k) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} w_{m,n}(k) \, . \, X_{i+m,j+n} + b^k \quad (2)$$

$Y_{i,j}(k)$ Is the output of the k[th] filter at position (i,j), $w_{m,n}(k)$ is the weight of the filter, $X_{i+m,j+n}$ is the input value at the shifted position. X is the input feature map, and b$^{(k)}$ is the bias term as represented in equation (3). The introduction of ReLU non-linearity and downsampling is performed at every layer through appropriate pooling operations. InceptionV3 backbone's final output is a feature map of 7 × 7 × 2048. The third stage does not directly feed this enhanced feature map into a purely classical Dense classifier. Instead, its feature representation is converted into a portable one-dimensional deep feature vector using global average pooling or flattening, followed by a feature compression layer. This step of compression is important since the dimensionality of the raw CNN feature vector is too large for quantum encoding. As a result, it can be said that the compressed feature vector is a low-dimensional but rich in semantics disease-specific pattern identifier. In the fourth stage, the compressed classical feature vector is mapped into quantum states using angle encoding. Let the compressed feature vector be represented as x = [x1,x2,x3,..., x$_n$] where each feature is encoded as a rotation angle of a quantum gate. The encoded quantum state is defined in equation(3).

$$| \psi(x) \rangle = \otimes_{i=1}^{n} Ry(xi) \, | 0 \rangle \quad (3)$$

where $R_y(x_i)$ represents the rotation gate applied to the i[th] qubit. After angle encoding, the quantum state is processed by a Variational Quantum Circuit (VQC) composed of parameterized gates and entanglement operations. This circuit modifies the encoded features and captures higher-order correlations among disease-specific features that may not be entirely learned by standard classical dense layers. The expectation values provided by the quantum circuit form the hybrid feature representation used for downstream classification. In the fifth stage, the quantum-transformed features are passed to a classical Dense classification head consisting of fully connected layers and a final Softmax output layer. The Softmax classifier computes the probability of each disease class according to:

$$P(y_{i)} = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \; for \; i = 1 \ldots k \quad (4)$$

where $z_i$is the logit corresponding to class? $i$, and $K$is the total number of output classes. The network is trained using the **categorical cross-entropy loss**, defined as:

$$L = -\sum_{i=1}^{K} y_i \log (\hat{y}_i) \quad (5)$$

where $y_i$ is the true label and $\hat{y}_i$ is the predicted probability for the class $i$. Dropout regularization is employed in the classical classification head to reduce overfitting and improve generalization.To preserve explainability, Grad-CAM is applied to the final convolutional layers of InceptionV3, generating visual heatmaps that indicate the regions of the leaf image most responsible for the prediction. Since the quantum module operates only on the compressed deep feature vector after CNN-based extraction, it does not affect the spatial feature maps required by Grad-CAM.

Therefore, the proposed TIGLeafNet-Q framework maintains the interpretability of the original TIGLeafNet while extending it with a hybrid quantum feature transformation module. Overall, the revised architecture integrates

standardized preprocessing, transfer learning, feature compression, angle-based quantum encoding, variational quantum feature transformation, classical classification, and Grad-CAM-based interpretability into a unified framework for robust and explainable tomato and cotton leaf disease detection. Figure 4 presents the full architecture of our proposed TIGLeafNet-Q model. It includes the InceptionV3 backbone, feature compression layer, angle encoding module, variational quantum circuit, classical dense classifier, and Grad-CAM visualization. The TIGLeafNet-Q model architecture accepts standard 224 × 224 × 3 RGB leaf images. These images are preprocessed using contrast enhancement, denoising, normalization, and augmentation. The preprocessed images are then passed through an InceptionV3 backbone to extract high-level visual features. The resulting feature vector is compressed into a lower-dimensional representation and encoded into quantum states through angle encoding. A variational quantum circuit then transforms the encoded states to model higher-order disease-specific feature interactions. The resulting quantum measurements are fed into classical dense layers for final disease classification. To preserve interpretability, Grad-CAM is applied to the final convolutional layers of InceptionV3 to highlight the image regions most responsible for each prediction. The selection of InceptionV3 is motivated by its proven ability to extract multi-scale features efficiently while maintaining computational feasibility. Feature compression is necessary to align high-dimensional CNN outputs with quantum circuit constraints. Angle encoding is chosen due to its efficiency in mapping classical data into quantum states with minimal circuit depth. Variational quantum circuits are employed to model non-linear feature interactions that classical dense layers may fail to capture. Grad-CAM is incorporated to ensure interpretability, which is critical for real-world agricultural applications where decision transparency is required.

### 4.3 Step-by-Step Procedure Overview

The step-by-step procedure is detailed in Figure 5 and outlined as follows.

Step 1: Image Resizing

All of the original tomato and cotton leaf images have been reduced to 224 x 224 pixels and converted from a single channel (grayscale) to a 3-channel RGB color space. The use of this standardized image size and type is important for compatibility with InceptionV3, as it needs to be fed with consistent dimension images as input: all images will have 224 x 224 x 3 dimensions. No other sizes will be accepted by the network in the last step of the training process.

Step 2: Image Preprocessing

Intermediate analysis may use Grayscale conversion as a preprocessing step; however, the RGB representation of the data is maintained in order to utilize this format as input to the final model. CLAHE is utilized to create an enhanced version of each pixel's intensity with respect to its surrounding pixels in an attempt to show even subtle lesions that have been masked by variable lighting conditions. The Gaussian filter smoothes the image with respect to high-frequency components of noise while maintaining significant structural features of the image. Each pixel value is then normalized based on the mean and standard deviation of ImageNet. For further processing purposes, binary thresholding may be used on grayscale versions of the image for purposes of visualization or auxiliary analysis; however, the primary input to the CNN will remain the normalized 224 x 224 x 3 image. Data augmentation techniques (e.g., random rotation, horizontal flip, zoom, etc.) increase the variability of the training set and improve the overall performance of the model.

Step 3: Labeling the Images

Following pre-processing, the images are labeled based on their respective disease type/class (e.g., Early blight, Late blight, Leaf mold) or as healthy. The use of folder organization and/or metadata allows for an organized and supervised labeling scheme for both tomato (healthy plus ten disease types/classes) and cotton (healthy plus three disease types/classes).

Step 4: Importing the Pre-trained Model and Setting Layers

InceptionV3, which has been pre-trained using ImageNet, was loaded with ImageNet and include_top=False. All 30 initial layers of convolutions were set to be non-trainable (trainable=False), while all the final 18 layers of convolutions were made trainable (trainable=True). A custom head is appended: Flatten → Dense(256, ReLU, Dropout 0.5) → Dense(128, ReLU, Dropout 0.5) → Dense(output_units, Softmax), where output_units

is 11 for tomato (10 diseases + 1 healthy) and 4 for cotton (3 diseases + 1 healthy). Categorical cross-entropy is used as the loss function, reflecting the multi-class nature of the problem.

Step 5: Feature Compression
The high-dimensional feature vector extracted from InceptionV3 is compressed using a dense projection layer to reduce redundancy and to match the qubit dimensionality required by the quantum circuit.

Step 6: Angle Encoding into Quantum States
The compressed classical features are encoded into quantum states using angle encoding, where each feature value determines the rotation angle of a quantum gate.

Step 7: Variational Quantum Circuit-Based Feature Transformation
The encoded quantum states are processed by a trainable variational quantum circuit composed of parameterized rotation gates and entanglement layers. This step captures higher-order correlations among disease-specific features.

Step 8: Hybrid Classification
The quantum measurement outputs are transferred back to classical dense layers, and the final disease category is predicted using a softmax classifier.

## 5. EXPERIMENTATION, RESULTS, AND ANALYSIS

### 5.1 Experimental Setup

The experiments were carried out in a Python-based development environment using Anaconda on a system configured with a 4 GB GPU, 16 GB RAM, and 1 TB storage. The proposed TIGLeafNet-Q framework was implemented using TensorFlow/Keras for the classical deep learning components and a quantum machine learning simulation framework for the variational quantum circuit layer. The classical InceptionV3 backbone was initialized with ImageNet pre-trained weights, while the hybrid quantum block was trained jointly with the compressed feature representation and the classical classification head.

InceptionV3 model, we trained on the tomato leaf recolonization dataset with over 22,980 images. All 22980 images were resized to size 224 × 224 × 3 for input to InceptionV3. The model was evaluated for performance and convergence under different epoch settings. In the last validation phase, the framework inquired about the results from the proposed model using a dataset. As such, the cotton leaf disease dataset was chosen. This dataset contains a total of 2204 images for training, validation, and testing. The outcomes showed the efficiency of the suggested architecture, reaching an accuracy of 97.23% on the cotton dataset at 60 epochs.

### 5.2 Performance Metrics

Accuracy measures how many of the predicted positive cases are actually correct. It indicates how well the model can correctly label a sample as positive, as defined in Equation (5).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (5)$$

TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives. Equation (6) shows that a low true positive rate or frequent misclassification of positive instances increases the denominator, consequently reducing precision.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (6)$$

The precision of a model is usually quite good since this method will identify many of the true positive cases. The approach will yield very few false positives as it has very few incorrectly identified positive results. Recall represents the proportion of actual positive cases that are predicted by the model. Recall indicates what portion of the true positive cases were actually predicted by the model, as demonstrated in equation (7).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (7)$$

When a model identifies all the positive cases perfectly, it can increase the model's recall. Sensitivity, or the true positive rate, is an alternative term for the rate at which a model successfully identifies positive cases. When a model has a high degree of sensitivity, it should be able to find most of the positive cases in the data set; however, it may not find all positive cases because there could be false negatives.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (8)$$

The measure of specificity (Equation 8) shows how well the model could correctly identify those samples that are actually free of a given disease, and therefore would be classified as negative for the disease, which should increase the amount of correct

negative predictions and reduce incorrect positive predictions.

### 5.3 Results

The experimental results demonstrate that the proposed TIGLeafNet-Q model is able to strongly classify diseases in both tomato and cotton datasets while retaining model interpretability. The purpose of a hybrid quantum feature transformation stage is to augment the feature representation beyond classical dense classifier capabilities. The results demonstrate that the proposed framework is able to support multi-crop disease recognition with good predictive accuracy and stable visual explanation. The proposed model leverages transfer learning to learn from 22980 images of tomato crop leaves categorized by disease, which includes Early Blight, Late Blight, and Leaf mold. As demonstrated in the figure, details pertaining to our image preprocessing steps are outlined in detail. We resized each image to 224×224, converted it to grayscale, applied CLAHE to enhance contrast, added Gaussian blur, normalized the pixels, and finally applied a binary thresholding operation, steps 5. To avoid overfitting our model to the training set, data augmentation techniques were also utilized, including rotating the images and horizontally flipping them. The model was trained using the Adam optimizer, categorical cross-entropy loss, a batch size of 32, and explicit training, validation, and test splits (18,345 / 4,585 / 50). After training with 60 epochs and early stopping, the model achieved a classification accuracy of 99.28% for tomato, demonstrating strong capability in distinguishing between diseased and healthy leaves. Performance metrics, including accuracy, precision, recall, F1-score, and loss, were computed from the test set confusion matrix, with precision, recall, and F1-score values to be reported class-wise or averaged, and final loss to be included if available.

In the second set of experiments conducted on the tomato plant leaf disease dataset, models were trained using different image resolutions and training epochs. When images of size 200 × 200 pixels were used, the model achieved an accuracy of 98.02% after 10 epochs and improved to 98.21% after 15 epochs. Using a higher image resolution of 224 × 224 pixels, the model obtained an accuracy of 98.10% at 10 epochs, which further increased significantly to 99.28% when trained for 60 epochs. These results demonstrate that increasing both image resolution and the number of training epochs contributes to improved classification performance. Figure 7a shows some example healthy leaf images, and Figure 7b shows the tomato plant leaf images with Bacterial spot, Late blight, and Mosaic viruses that were detected by our proposed model. The highlighted spots of the leaf are highly affected by the following kinds of diseases: Tomato bacterial spot is induced by the microorganisms Xanthomonas gardneri, Xanthomonas euvesicatoria, Xanthomonas aspera, and Xanthomonas vesicatoria. These bacterial infections, which can cause symptoms or not, can enter a garden on contaminated seeds and transplants. Figure 6 shows some examples of disease-affected areas in tomato leaf images.

The pathogen that causes tomato late blight, Phytophthora infestans, requires tissue to survive. A plant pathogenic virus is the tomato mosaic virus, ToMV. It affects many different plants, including tomatoes and their leaves, and is present everywhere. The bacteria can attack the stems, leaves, bracts, and bolls of the cotton plant at any stage of its growth. It results in boll rot, black vein, black arm seedling blight, and leaf spot. Cotyledons develop small, green, spherical, water-soaked spots that eventually turn brown. Leaf miners are little, 2mm long, greyish black flies that feed on the underside of leaves. The feeding activity damages healthy leaf tissue, reducing the plant's ability to absorb sufficient sunlight and often resulting in disease, as illustrated in Figure 7.

In machine learning, the loss value is a key metric that quantifies how well a model's predictions match the actual outcomes, and it plays a central role in evaluating model performance. During training, loss is computed after each batch or epoch of data is processed, and ideally, it should decrease over time, indicating that the model is improving its accuracy. An epoch is one complete pass through the training data and an iteration is a single update to the model's parameters after processing a data batch As the model trains the loss ideally goes down showing that its predictions are getting closer to the actual values After training and fixing the model's parameters its correctness is assessed using performance metrics like accuracy precision recall and F1-score These metrics assess the model's performance on novel, unencountered data. The training loss indicates learning progress, but post-training metrics better reflect real-world efficacy.

Figure 8 shows the training and validation accuracy and loss curves for the proposed model on the tomato leaf dataset. We resized images to 224x224 pixels and conducted training for 15 epochs. The proposed model's performance is illustrated by its learning curve in Figure 9. In this figure, we can see that both training accuracy and validation accuracy improved significantly as a result of the model's ability to

improve its predictions. The proposed model was able to reach 96.5% training accuracy and 99.7% validation accuracy. The loss of the model during training (train loss) started at 0.3 and ended with a value of 3.5, while the loss of the model on the validation set (validation loss) started at 0.25 and ended with a value of 2.32. In Fig. 9, we present the results of the proposed model for the validation and training accuracy and loss on the tomato leaf dataset for an image size of 200 x 200 pixels and after 15 epochs. The training accuracy starts at 10% and ends at 99%, and the validation accuracy ranges from 35% to 99.9%. The train loss starts at 0.2 and ends at 2.6, while the validation loss starts at 0.125 and ends at 2.1. The model receives feedback from the test samples again after comparing the recorded number of faults (0–1 losses) to the actual targets. Next, the misclassification rate is determined. Figures 10 and 11 show the proposed model's training, validation, and precision results for image sizes (224 x 224) and (200 x 200) on the tomato leaf dataset. The train recall starts at 9% and ends at 97%; the validation recall starts at 13% and ends at 98.2% on the tomato leaf dataset. The train's precision accuracy starts at 4% and ends at 94.2%, and the validation precision starts at 53.2% and ends at 96.4%. As shown in the figure12, our model produced loss values of 0.0344, 0.0348, 0.0095, 0.0927, 0.0186, 0.1278, 0.0356, 0.0852,0.0511,0.0879, 0.1663, 0.1418, 0.0397 and accuracy values of 0.9957, 0.9915, 0.9745, 0.9915, 0.9830, 0.9872, 0.9617, 0.9702, 0.9787, 0.9787 and validation loss of 0.0121, 1.266, 0.0015, 0.0083, 8.302, 0.2074, 0.0035, 0.0457, 0.1151 and validation accuracy of 0.9643, 0.9821, 0.9464, 0.9643, for 60 epoch. The training loss begins at 0.0111 and increases to a peak of 8.918, while the validation loss starts at zero and reaches 8.834 by the end.

The proposed model's training, validation, sensitivity, and specificity results on the tomato leaf dataset are shown in Figure 13. The train sensitivity starts at 61.3% and ends at 97%, while the validation sensitivity starts at 84.2% and ends at 99.5%. The train specificity starts at 65.2% and ends at 96.9%, while the validation sensitivity starts at 72.42% and ends at 99.2%. The second dataset consists of 2204 images, of which 1951 are training images and 253 are test images, which are both considered training and validation images. Bacterial blight, bacterial blight with leaf miner, and healthy cotton leaf image are the classes available in the dataset. As shown in Figure 14, the model demonstrated consistent training and validation performance over 10 epochs. Training loss ranged from 0.21 to 0.51, while validation loss varied between 0.48 and 1.52. Accuracy values remained high, with training accuracy reaching up to 0.94 and validation accuracy peaking at 0.93. Table 3 presents a comparative overview of current methodologies and their documented performance. Our transfer learning model outperformed existing approaches, achieving 99.28% accuracy on a 22,980-image tomato leaf dataset and 97.23% on a 2,204-image dataset within 60 epochs. Table 3 shows that the proposed TIGLeafNet model outperforms other existing techniques in comparative analysis. Before our study, the leading benchmark method, which reached 98.63% accuracy, was k-FLBPCM paired with SVM [40] from a selection of 23 methods. These included established machine learning techniques like SVM, GLCM with SVM, and DLQP with SVM, alongside deep learning models such as VGG [31], ResNet152V2 [37], R-CNN [33], and DCNN [34]. While prior work, including DCNN [34] at 98.40%, Improved SVM [46] at 98.3%, and CNN-based frameworks [12], [47] at 98%, demonstrated effectiveness, our approach achieved higher performance. TIGLeafNet advanced performance significantly, reaching 99.28% accuracy for tomatoes and 97.23% for cotton. As shown in figure 15 ,The model performs reliably across a range of crops, demonstrating its broad applicability and accurate disease classification. On the other hand, validation loss is pretty volatile, presenting a few sudden spikes (e.g., at the 5th and 10th epoch), spiking on the level of 1.6. This is indicative of potential overfitting and/or a lack of robustness in the model's ability to generalize to new data. Similarly, as the training accuracy remains high (90-95%) throughout all the epochs, the validation accuracy fluctuates a lot, going below 90% in some epochs. These kinds of discrepancies between training and validation scores would tend to indicate an over-fit model that does not generalize very well. Some strategies, like adding regularizers (eg, L2 or dropout), data augmentation, or other validation-centric strategies, should be applied to overcome this. The training and validation accuracy and loss results of the proposed method on the cotton leaf image dataset are shown in Figure 16 over 20 epochs. The training loss, which ranges from 0.3 to 0.5, remains low and modest throughout the iterations, suggesting that the learning from training data is good. It is a contrast to the validation loss, which starts at about 2.0 and already after a few epochs goes down, which means that there is some improvement in generalization. It does fluctuate in the later epochs, and has several sharp peaks (something like at epochs 13 and 17), which might mean it is not stable and susceptible to

overfitting. The training accuracy remains consistently high at over 96% and at a peak of around 98%, indicating good performance on the training dataset. And the validation accuracy begins by rising from about 88% to 94%, then falls again, and keeps falling at the end of the training to below 90%. Figure 17 presents the training and validation performance of the model over 50 epochs. The left graph displays the training and validation loss, where the training loss remains consistently low and stable, indicating that the model is learning effectively from the training data. The validation loss's significant fluctuations throughout epochs indicate potential overfitting or inconsistent generalization. The graph on the right displays the training and validation accuracy. Training accuracy steadily improves, reaching almost 99%, while validation accuracy is inconsistent and lower. This finding suggests that the model fits the training set well but may need to be regularized or validated on a more diverse data set to help it generalize better.

As illustrated in Figure 18, the proposed model achieved training loss values ranging from 0.1320 to 2.5929 and accuracy up to 0.9646. Validation loss varied between 0.3649 and 2.0528, with validation accuracy reaching as high as 0.9605 across 60 epochs. The proposed model is applied to the cotton leaf disease image dataset, and it has produced the following values as accuracy: 92.8% of accuracy for 10 epochs, 94.7 % of accuracy for 15 epochs, 94.86 % of accuracy for 20 epochs, 96.05% of accuracy for 50 epochs, and 97.23 % of accuracy for 60 epochs, as shown in figure 19.

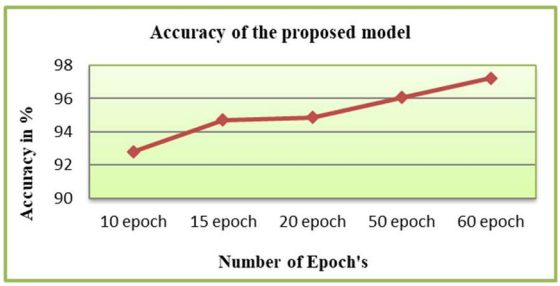



*Figure 19: Proposed model's accuracy for various epochs on the cotton leaf dataset*

### 5.4 Disease Region Localization using Grad-CAM

In the proposed TIGLeafNet-Q framework, Grad-CAM continues to provide spatial interpretability because the explainability mechanism is applied to the convolutional feature maps generated by InceptionV3. The quantum module operates on compressed feature vectors after classical feature extraction and therefore does not alter the spatial activation maps required for Grad-CAM computation. As a result, explainability is preserved in the hybrid architecture. To clarify our tomato leaf disease classification, Grad-CAM highlighted disease-affected regions. This method highlights those areas of the image that most influenced the model's decision. By placing these activation maps on the original image, the model can be evaluated and verified by any expert. The heatmap from Figure 20 visually represents which parts of the leaf disease detection models are activated. The reds, yellows, and pinks represent high activation areas indicating the possible presence of disease. In comparison, the blues and greens indicate low activation in the healthy regions of the leaf. Next, we precisely segmented these high activation regions in the image to demarcate the most prominent disease areas using bounding boxes or contours. This tool precisely maps infected leaf areas. This enables direct visual comparison between the predicted model and observed symptoms. The improved accuracy and clarity of the model are useful for practical agricultural purposes. Researchers can find a simple explanation of how the model makes its decision, and farmers and agriculture experts get a clear picture of disease symptoms and how a disease spreads. By looking at multiple Grad-CAM outputs, we could identify any bias that might exist in the model as well as atypical disease manifestations, which would help us fine-tune the data and improve the accuracy.

### 5.5 Analysis

Table 3 compares the existing methods and their performance results. With our transfer learning model, we achieved an accuracy of 99.28% using a 22,980-image dataset of tomato leaves and 97.23% using a 2,204-image dataset within 60 epochs, thus outperforming existing approaches. As demonstrated by Table 3, the proposed TIGLeafNet model outperformed the other existing comparative techniques. Prior to our study, the benchmark method with the highest accuracy (98.63% accuracy) of the study was k-FLBPCM-SVM (associated with 23 methods) [40]. Some of these are established machine learning methods, such as SVM, GLCM with SVM, and DLQP with SVM, along with deep learning models such as VGG [31], ResNet152V2 [37], R-CNN [33], and DCNN [34]. Previous work achieved an effectiveness of DCNN [34] 98.40%, Improved SVM [46] 98.3% CNN based framework [12], and [47] 98%. However, we achieve a better performance. TIGLeafNet improves performance considerably: 99.28% for tomato

recognition and 97.23% for cotton. The model successfully works on several crops, which denotes its wide adaptability and accurate disease prediction.

*Table 3. Accuracy Of Proposed And Existing Models*

| S. No | Techniques or Methods | Performance metrics (Accuracy) |
|---|---|---|
| 1 | VGG [31] | 93.3% |
| 2 | Deep Learning [32] | 96.5% |
| 3 | R-CNN [33] | 96.7% |
| 4 | DCNN [34] | 98.40% |
| 5 | Deep CNN [35] | 91% |
| 6 | MDCNN [36] | 83.9% |
| 7 | Pre-trained ResNet152V2[37] | 97% |
| 8 | SVM &CNN [38] | 88% & 96.7% |
| 9 | Lightweight CNN[39] | 87.6% |
| 10 | k-FLBPCM along with SVM[40] | 98.63% |
| 11 | DLQP method with SVM[41] | 96.53% |
| 12 | LBP algorithm with SVM[42] | 95% |
| 13 | GLCM method with SVM [43] | 98.5% |
| 14 | SVM [44] | 92% |
| 15 | DL-based framework using LeNet [45] | 94.8% |
| 16 | CNN-based framework[12] | 98% |
| 17 | Improved SVM[46] | 98.3% |
| 18 | AlexNet & DenseNet201 with SVM[24] | 97.56% |
| 19 | CNN[28] | 91.25% |
| 20 | CNN[47] | 98% |
| 21 | Hybrid QViT-based model [48] | 98.5% |
| 22 | Quantum convolutional neural network (QCNN) [50] | 96.9% |
| 23 | QCNN-based model[51] | 94.96% |
| 24 | Proposed model on Tomato leaf dataset | 99.28% |
| 25 | Proposed model on Cotton leaf dataset | 97.23% |

According to Figure 21, the performance of the TIGLeafNet model is better than that of other techniques for the tomato leaf dataset. Deep learning architectures like VGG (31) achieved 93.3%. Deep Learning (32) showed 96.5%, R-CNN (33) with 96.7%, DCNN (34) gained an accuracy of 98.40%, and Pre-trained ResNet152V2 (37), which attained 97%, shows similarity in accuracy among the existing methods. CNN architectures from [12] and [47] achieved an accuracy of 98%, which is greater than the LBP-SVM, which achieved 95%, and a joint SVM-CNN approach, which achieved 96.7%. Deep CNN [35] and CNN [28], other CNN variants, were able to achieve an accuracy of 91% and 91.25%, respectively. TIGLeafNet's accuracy on the tomato leaf dataset is 99.28%, much greater than other methods. This indicates that the model is quite effective in detecting and classifying tomato leaf diseases more accurately than traditional and deep learning models.

TIGLeafNet achieves better performance compared to current techniques on various leaf datasets, as shown in Figure 22. Recent CNN and SVM approaches have relatively low performance. The k-fold local fuzzy pattern-based complex models (k-FLBPCM) and GLCM with SVM give a strong performance of 98.63% and 98.5%, respectively. The proposed TIGLeafNet (Tomato) outperforms QCNN [50] and QCNN-based model [51] by 2.38% and 4.32%, respectively. Similarly, TIGLeafNet (Cotton) shows improvements of 0.33% over QCNN and 2.27% over the QCNN-based model, confirming its superior performance. However, they do not surpass the model proposed in this study. TIGLeafNet performs better than all other techniques at classifying plant diseases across various crops, achieving accuracy rates of 99.28% on Tomato leaves and 97.23% on Cotton leaves. According to the scientists who developed it, it was tested on several categories of deformed cotton leaves. They were included among other types of images to ensure validation of the effectiveness of the computer vision task for the multi-class CNN architecture. The proposed technique, which incorporates transfer learning, quantum feature transformation, and Grad-CAM-based explanation, provides a scalable and explainable solution for automating the plant disease identification procedure.

## 6. DIFFERENCE FROM PRIOR WORK

The proposed TIGLeafNet-Q framework distinguishes itself from existing approaches through several key innovations. Unlike purely classical

convolutional neural network models, the proposed method integrates hybrid quantum–classical learning to enable enhanced feature representation, allowing the model to capture complex higher-order correlations that are often difficult to model using conventional deep learning techniques. Furthermore, while most existing studies focus on single-crop disease detection, TIGLeafNet-Q demonstrates effective generalization across multiple crops, specifically tomato and cotton datasets, thereby improving its applicability in real-world agricultural scenarios. In contrast to many hybrid quantum models that overlook interpretability, the proposed framework incorporates Grad-CAM to preserve spatial explainability by highlighting disease-relevant regions in leaf images. Additionally, the architecture combines transfer learning with quantum feature encoding, ensuring both computational efficiency and improved representational capability. As a result, the proposed model offers several advantages, including high classification accuracy, robustness under varying conditions, enhanced modeling of feature interactions, and interpretable predictions suitable for practical agricultural decision-making systems.

## 6. CONCLUSION AND FUTURE SCOPE

This study introduced TIGLeafNet-Q, a hybrid quantum–classical explainable framework designed to address key limitations in plant disease detection, including poor generalization, inadequate higher-order feature representation, and lack of interpretability. The proposed framework integrates InceptionV3-based transfer learning with feature compression, angle encoding, and a variational quantum circuit to enhance feature transformation and classification performance. Furthermore, the incorporation of Grad-CAM ensures model transparency by highlighting disease-relevant regions in leaf images. Experimental results on tomato and cotton datasets demonstrate that the proposed model achieves high classification accuracy while maintaining robustness and interpretability, thereby validating its effectiveness as a multi-crop disease detection solution. The integration of quantum feature transformation with classical deep learning significantly improves the model's ability to capture complex feature interactions compared to conventional approaches. However, the study is limited by the use of simulated quantum circuits and the presence of validation fluctuations, indicating potential overfitting and constraints in generalization. Future work will focus on deploying the framework on real quantum hardware, enhancing generalization through advanced regularization strategies, and extending the model to additional crop species and real-world agricultural environments. These directions will further strengthen the applicability of hybrid quantum–classical learning in precision agriculture systems.

**Declarations:**

**Human Ethics and Consent to Participate declarations:** Not applicable

**Competing interests:** The authors declare no competing interests.

**Author contributions:**

DV: Conceptualization, Methodology, Software, Writing – Original manuscript; KT: Study, Data curation and formal data analysis, Writing – Original manuscript; VSJ: Data collection, Methodology, Validation, Visualization; AB: Investigation, Visualization; VG: Writing the Paper, Supervision, Validation; MS: Methodology, Visualization; ASN: Literature review, Supervision, Review & Revise – Original manuscript;

## REFERENCES:

[1] Zhao, S., Peng, Y., Liu, J., & Wu, S. (2021). Tomato leaf disease diagnosis based on improved convolution neural network by attention module. Agriculture, 11(7), 651.

[2] Djimeli-Tsajio, A. B., Thierry, N., Jean-Pierre, L. T., Kapche, T. F., & Nagabhushan, P. (2022). Improved detection and identification approach in tomato leaf disease using transformation and combination of transfer learning features. Journal of Plant Diseases and Protection, 129(3), 665-674.

[3] Khandare, P. M., Dhutraj, D. N., & Ambadkar, C. V. (2019). Epidemiological Studies of Tomato Leaf Curl Virus in Marathwada Region of Maharashtra, India. Int. J. Curr. Microbiol. App. Sci, 8(11), 688-697.

[4] Rishi, N. (2012). SN Dasgupta Memorial Lecture Award (2003)-Current status of begomoviruses in the Indian subcontinent-NARAYAN RISHI. Indian Phytopathology, 57(4), 396-407.

[5] Rojas, M. R., Hagen, C., Lucas, W. J., & Gilbertson, R. L. (2005). Exploiting chinks in the plant's armor: evolution and emergence of geminiviruses. Annu. Rev. Phytopathol., 43, 361-394.

[6] Polston, J. E., McGovern, R. J., & Brown, L. G. (1999). Introduction of tomato yellow leaf curl virus in Florida and implications for the spread of this and other geminiviruses of tomato. Plant Disease, 83(11), 984-988.

[7] Ashwathappa, K. V., Venkataravanappa, V., Reddy, C. L., & Reddy, M. K. (2020). Association of Tomato leaf curl New Delhi virus with mosaic and leaf curl disease of Chrysanthemum and its whitefly cryptic species. Indian Phytopathology, 73, 533-542.

[8] Li, L., Zhang, S., & Wang, B. (2021). Plant disease detection and classification by deep learning—a review. IEEE Access, 9, 56683-56698.

[9] Roy, A., Spoorthi, P., Panwar, G., Bag, M. K., Prasad, T. V., Kumar, G., ... & Dutta, M. (2013). Molecular evidence for the occurrence of tomato leaf curl New Delhi virus in ash gourd (Benincasa hispida) germplasm showing a severe yellow stunt disease in India. Indian Journal of Virology, 24, 74-77.

[10] Kim, N., Kim, J., Bang, B., Kim, I., Lee, H. H., Park, J., & Seo, Y. S. (2016). Comparative analyses of Tomato yellow leaf curl virus C4 protein-interacting host proteins in healthy and infected tomato tissues. The plant pathology journal, 32(5), 377.

[11] Picó, B., Díez, M. J., & Nuez, F. (1998). Evaluation of whitefly-mediated inoculation techniques to screen Lycopersicon esculentum and wild relatives for resistance to Tomato yellow leaf curl virus. Euphytica, 101, 259-271.

[12]Chakraborty, S., Pandey, P. K., Banerjee, M. K., Kalloo, G., & Fauquet, C. M. (2003). Tomato leaf curl Gujarat virus, a new begomovirus species causing a severe leaf curl disease of tomato in Varanasi, India. Phytopathology, 93(12), 1485-1495.

[13]Thangaraj, R., Anandamurugan, S., & Kaliappan, V. K. (2021). Automated tomato leaf disease classification using transfer learning-based deep convolutional neural network. Journal of Plant Diseases and Protection, 128(1), 73-86.

[14]Reem, A., Almezgagi, M., Al-Shaebi, F., Al-Shehari, W. A., & Kumal, J. P. P. (2020). Application of Rolling Circle Amplification (RCA) to Detect the Pathogens of Infectious Diseases. Infect Dis Diag Treat, 4(158), 2577-1515.

[15]De Luna, R. G., Baldovino, R. G., Cotoco, E. A., De Ocampo, A. L. P., Valenzuela, I. C., Culaba, A. B., & Gokongwei, E. P. D. (2017, December). Identification of Philippine herbal medicine plant leaf using an artificial neural network. In 2017, IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) (pp. 1-8). IEEE.

[16]Kamaal, N., Akram, M., & Agnihotri, A. K. (2015). Molecular Evidence for the Association of Tomato leaf curl G ujarat virus with a Leaf Curl Disease of P haseolus vulgaris L. Journal of Phytopathology, 163(1), 58-62.

[17]Chakraborty, S. (2009). Tomato leaf curl viruses from India. In Desk encyclopedia of plant and fungal virology (pp. 339-347). United Kingdom: Elsevier.

[18]Sattar, M. N., Kvarnheden, A., Saeed, M., & Briddon, R. W. (2013). Cotton leaf curl disease–an emerging threat to cotton production worldwide. Journal of General Virology, 94(4), 695-710.

[19]De Luna, R. G., Dadios, E. P., & Bandala, A. A. (2018, October). Automated image capturing system for deep learning-based tomato plant leaf disease detection and recognition. In TENCON 2018-2018 IEEE Region 10 Conference (pp. 1414-1419). IEEE.

[20]Friedman, M. (2013). Anticarcinogenic, cardioprotective, and other health benefits of tomato compounds lycopene, α-tomatine, and tomatidine in pure form and in fresh and processed tomatoes. Journal of agricultural and food chemistry, 61(40), 9534-9550.

[21]Cohen, S., & Antignus, Y. (1994). Tomato yellow leaf curl virus, a whitefly-borne geminivirus of tomatoes. Advances in disease vector research, 259-288.

[22] Kushwaha, N., Singh, A. K., Chattopadhyay, B., & Chakraborty, S. (2010). Recent advances in geminivirus detection and future perspectives. J Plant Prot Sci, 2, 1-18.

[23]Kushwaha, N., Singh, A. K., Basu, S., & Chakraborty, S. (2015). Differential response of diverse solanaceous hosts to tomato leaf curl New Delhi virus infection indicates coordinated action of NBS-LRR and RNAi-mediated host defense. Archives of Virology, 160, 1499-1509.

[24]Turkoglu, M., Yanikoğlu, B., & Hanbay, D. (2022). PlantDiseaseNet: Convolutional neural network ensemble for plant disease and pest detection. Signal, Image and Video Processing, 16(2), 301-309.

[25]Azath, M., Zekiwos, M., & Bruck, A. (2021). Research Article Deep Learning-Based Image

Processing for Cotton Leaf Disease and Pest Diagnosis.

[26] Albattah, W., Nawaz, M., Javed, A., Masood, M., & Albahli, S. (2022). A novel deep learning method for detection and classification of plant diseases. Complex & Intelligent Systems, 1-18.

[27] Pantazi, X. E., Moshou, D., & Tamouridou, A. A. (2019). Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers. Computers and electronics in agriculture, 156, 96-104.

[28] Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato leaf disease detection using convolution neural Polston network. Procedia Computer Science, 167, 293-301.

[29]Abulizi, A., Ye, J., Abudukelimu, H., & Guo, W. (2025). DM-YOLO: Improved YOLOv9 model for tomato leaf disease detection. Frontiers in Plant Science, 15, 1473928.

[30]Osmenaj, Z., Tseliki, E. M., Kapellaki, S. H., Tselikis, G., & Tselikas, N. D. (2025). From Pixels to Diagnosis: Implementing and Evaluating a CNN Model for Tomato Leaf Disease Detection. Information, 16(3), 231.

[31]Vipinadas, M. J., & Thamizharasi, A. (2016). Detection and Grading of diseases in Banana leaves using Machine Learning. International Journal of Scientific & Engineering Research, 7(7), 916-924.

[32]Mahamud, S. S., Ayve, K. N., Uddin, A. H., & Arif, A. S. M. (2022). Tomato Leaf Disease Recognition with Deep Transfer Learning. In Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2022, Volume 2 (pp. 203-211). Singapore: Springer Nature Singapore.

[33]Nagamani, H. S., & Sarojadevi, H. (2022). Tomato leaf disease detection using deep learning techniques. International Journal of Advanced Computer Science and Applications, 13(1).

[34]Selvaraj, M. G., Vergara, A., Ruiz, H., Safari, N., Elayabalan, S., Ocimati, W., & Blomme, G. (2019). AI-powered banana disease and pest detection. Plant Methods, 15, 1-11.

[35]Anandhakrishnan, T., & Jaisakthi, S. M. (2022). Deep Convolutional Neural Networks for image based tomato leaf disease detection. Sustainable Chemistry and Pharmacy, 30, 100793.

[36]Ramcharan, A., McCloskey, P., Baranowski, K., Mbilinyi, N., Mrisho, L., Ndalahwa, M., ... & Hughes, D. P. (2019). A mobile-based deep learning model for cassava disease diagnosis. Frontiers in plant science, 272.

[37] Djimeli-Tsajio, A. B., Thierry, N., Jean-Pierre, L. T., Kapche, T. F., & Nagabhushan, P. (2022). Improved detection and identification approach in tomato leaf disease using transformation and combination of transfer learning features. Journal of Plant Diseases and Protection, 129(3), 665-674.

[38]Harakannanavar, S. S., Rudagi, J. M., Puranikmath, V. I., Siddiqua, A., & Pramodhini, R. (2022). Plant leaf disease detection using computer vision and machine learning algorithms. Global Transitions Proceedings, 3(1), 305-310.

[39]Li, X., Li, X., Zhang, S., Zhang, G., Zhang, M., & Shang, H. (2022). SLViT: Shuffle-convolution-based lightweight Vision transformer for effective diagnosis of sugarcane leaf diseases. Journal of King Saud University-Computer and Information Sciences.

[40] Le, V. N. T., Ahderom, S., Apopei, B., & Alameh, K. (2020). A novel method for detecting morphologically similar crops and weeds based on the combination of contour masks and filtered Local Binary Pattern operators. GigaScience, 9(3), giaa017.

[41]Ahmad, W., Shah, S. M., & Irtaza, A. (2020). Plants disease phenotyping using quinary patterns as texture descriptor. KSII Transactions on Internet and Information Systems (TIIS), 14(8), 3312-3327.

[43]Kuricheti, G., & Supriya, P. (2019, April). Computer vision based turmeric leaf disease detection and classification: a step to smart agriculture. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 545-549). IEEE.

[44]Vipinadas, M. J., & Thamizharasi, A. (2016). Detection and Grading of diseases in Banana leaves using Machine Learning. International Journal of Scientific & Engineering Research, 7(7), 916-924.

[45]Albattah, W., Nawaz, M., Javed, A., Masood, M., & Albahli, S. (2022). A novel deep learning method for the detection and classification of plant diseases. Complex & Intelligent Systems, 1-18.

[46]Ansari, A. S., Jawarneh, M., Ritonga, M., Jamwal, P., Mohammadi, M. S., Veluri, R. K., ... & Shah, M. A. (2022). Improved Support Vector Machine and Image Processing Enabled Methodology for Detection and Classification of

Grape Leaf Disease. Journal of Food Quality, 2022.

[47] Osmenaj, Z., Tseliki, E. M., Kapellaki, S. H., Tselikis, G., & Tselikas, N. D. (2025). From Pixels to Diagnosis: Implementing and Evaluating a CNN Model for Tomato Leaf Disease Detection. Information, 16(3), 231.

[48] Navya, D., & Rao, M. U. A quantum-enhanced vision transformer framework with hybrid optimization for efficient tomato leaf disease detection. In Artificial Intelligence, Computational Intelligence and Inclusive Technologies (pp. 728-739). CRC Press.

[49] Nivethitha T, Venkataramanan C. A Quantum-Enhanced Deep Learning Framework for Precise Leaf Disease Detection in Tomato and Cotton Crops. Iranian Journal of Science and Technology, Transactions of Electrical Engineering. 2025 Oct 3:1-32.

[50] Genemo, M., 2023, August. Quantum convolutional neural network for agricultural mechanization and plant disease detection. In International Conference on Image Processing and Capsule Networks (pp. 225-237). Singapore: Springer Nature Singapore.

[51] Jhansi, G., Sudhakar, P., Surya, C.C., Kumar, A.K., Chunduru, M.R., and Satyanarayana, G., 2025, October. Quantum Convolutional Neural Network Framework for Efficient and Accurate Plant Disease Diagnosis. In 2025 2nd International Conference on Electronic Circuits and Signaling Technologies (ICECST) (pp. 934-939). IEEE.

[52]Sandhya, M., Senthilraja, G., Priyadharshini, E., Rani, L.U., Harideekshayini, R., Nishanthi, M., Anand, T. and Subramanian, K.S., 2025. Early detection of plant diseases and their management using quantum dots: Status and strategies. Journal of Fluorescence, 35(11), pp.10945-10960.

*Table 1: Literature On Plant Leaf Disease Detection And Classification*

| S. No | Techniques or Methods | Performance metrics (Accuracy) | Advantages | Gap identified |
|---|---|---|---|---|
| 1 | VGG [31] | 93.30% | Automatic identification of leaf disease | Considered only three categories of diseases |
| 2 | Deep Learning [32] | 96.50% | The model produced good accuracy | Lack of validation on real-world field data with varying environmental conditions. |
| 3 | R-CNN [33] | 96.70% | The model achieved good accuracy | Limited Generalization Due to Small |
| 4 | DCNN [34] | 98.40% | Mobile app | Not applicable for multiple diseases |
| 5 | Deep CNN [35] | 91% | The model achieved good accuracy | Lack of Real-World Generalizatio |
| 6 | MDCNN [36] | 83.90% | Mobile app | Lower accuracy, and only three types of diseases are considered for testing. |
| 7 | Pre-trained ResNet152V2[37] | 97% | The model achieved good accuracy | Absence of fine-tuning might lead to overfitting in the model. |
| 8 | SVM &CNN [38] | 88% & 96.7% | The model produced good accuracy | Only 600 samples of tomato leaf are considered for evaluation and testing. |
| 9 | Lightweight CNN[39] | 87.60% | Enabled a two-dimensional vector | Over noisy inputs, classification efficiency decreases. |
| 10 | k-FLBPCM along with SVM[40] | 98.63% | Improve the classification of plants with comparable morphological textures. | The accuracy of detection decreases for the distorted samples. |
| 11 | DLQP method with SVM[41] | 96.53% | classify plant leaf diseases despite extreme scale and angle changes in the input samples | The accuracy of detection decreases for the distorted samples |
| 12 | LBP algorithm with SVM[42] | 95% | The model has increased generalization ability. | Over noisy inputs, classification efficiency decreases. |
| 13 | GLCM method wi SVM [43] | 98.50% | Detect the disease-affected leaf from the background noise | This model has a high computational cost |
| 14 | SVM [44] | 92% | This method can locate a portion of the disease leaf from the blurry input image. | Classification performance decreases for samples having extreme brightness changes. |
| 15 | DL-based framewo using LeNet [45] | 94.80% | This model needs less training data. | This approach performs poorly when processing images with noise. |
| 16 | CNN-based framework[12] | 98% | This model works efficiently for noisy samples. | This approach is costly in terms computing. |

| | | | | |
|---|---|---|---|---|
| 17 | Improved SVM[46] | 98.30% | The model produced good accuracy | This model is not robust to situations in real life and has only been tested for the categorization of grape leaf disease, and it used only 400 images for experimentation purposes. |
| 18 | AlexNet & DenseNet201 with SVM[24] | 97.56% | This model is robust to leaf disease classification when light variations are present. | High computational cost |
| 19 | CNN[28] | 91.25% | The model achieved good accuracy | The model may overfit due to class imbalance and lacks validation on multi-modal inputs, limiting generalization. Additionally, interpretability is hindered by insufficient fine-grained annotations for complex cases. |
| 20 | CNN[47] | 98% | The model achieved good accuracy | Poor generalization |

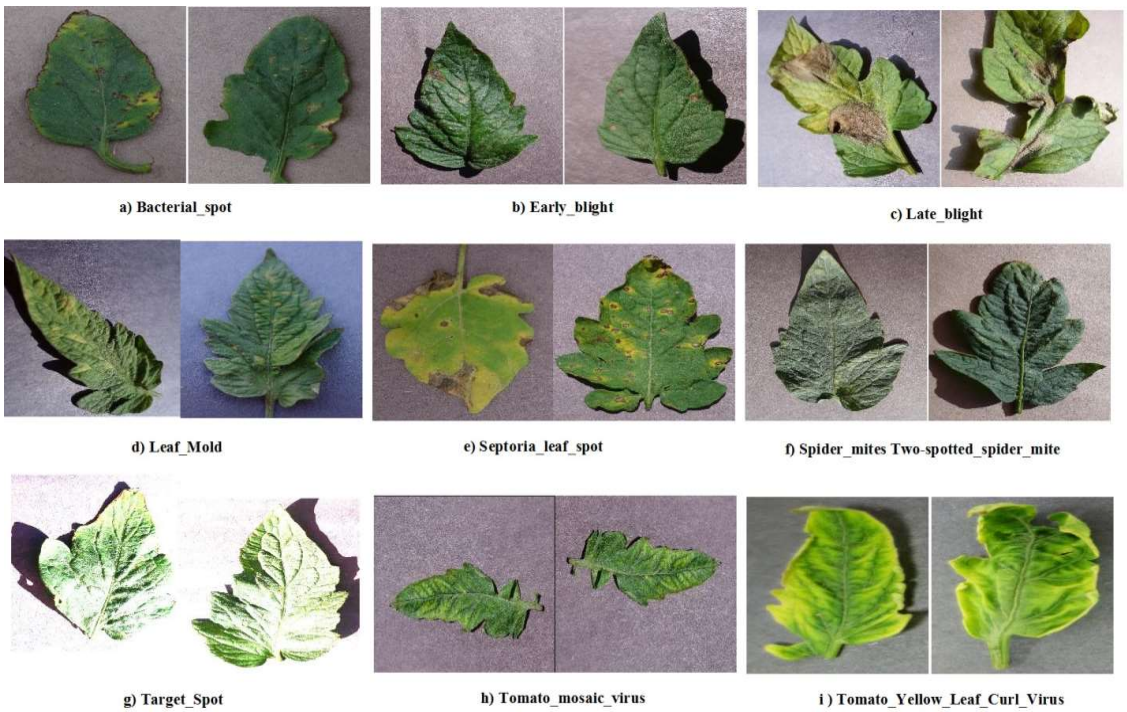



***Figure 1:*** *Different categories of tomato plant leaf disease images*

Journal of Theoretical and Applied Information Technology
30th April 2026. Vol.104. No.8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

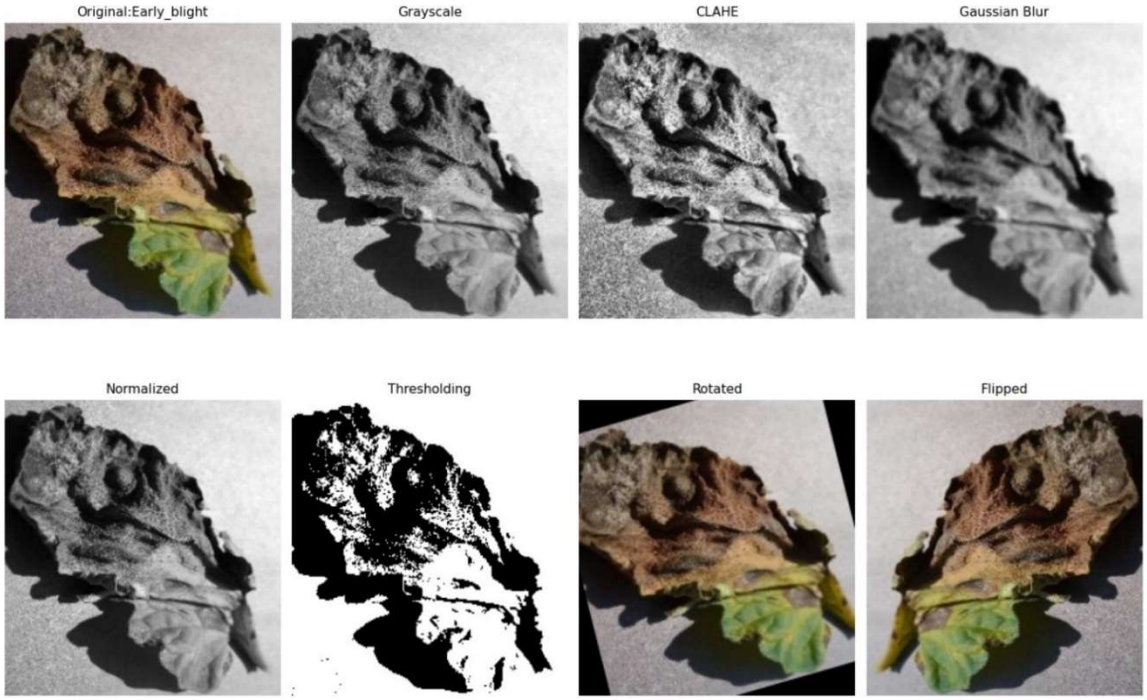



***Figure 2:*** *Examples of Preprocessed Images*


***Figure 3:*** *Resultant Image after Preprocessing*

Journal of Theoretical and Applied Information Technology
30th April 2026. Vol.104. No.8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

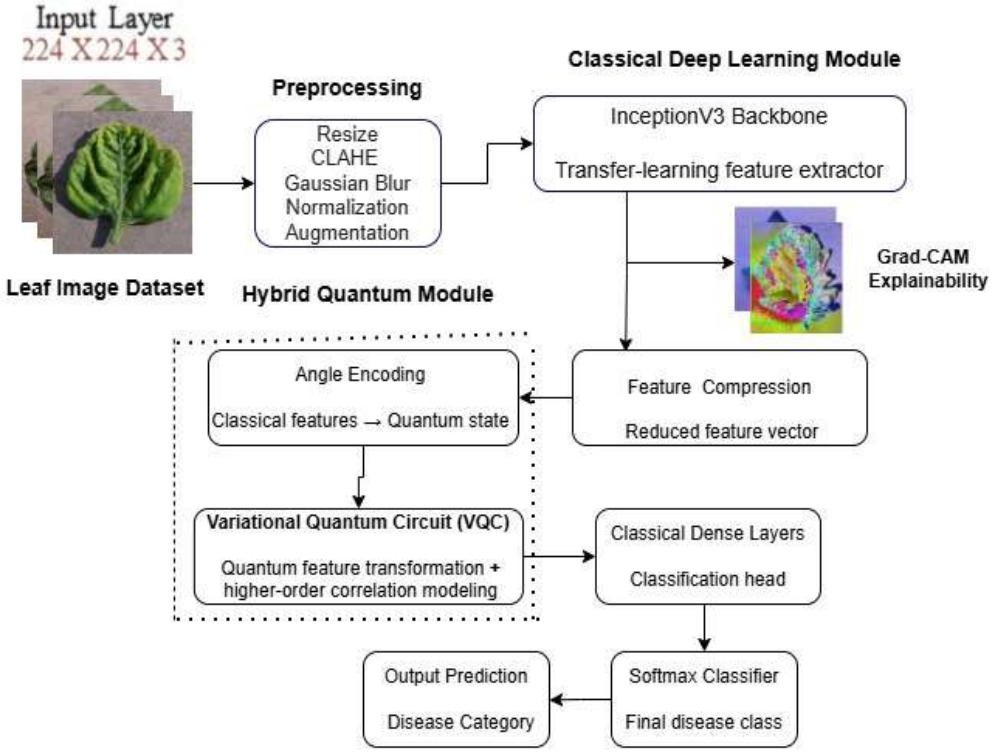



***Figure 4:*** *Proposed TIGLeafNet-Q architecture with hybrid quantum feature transformation and Grad-CAM explainability*

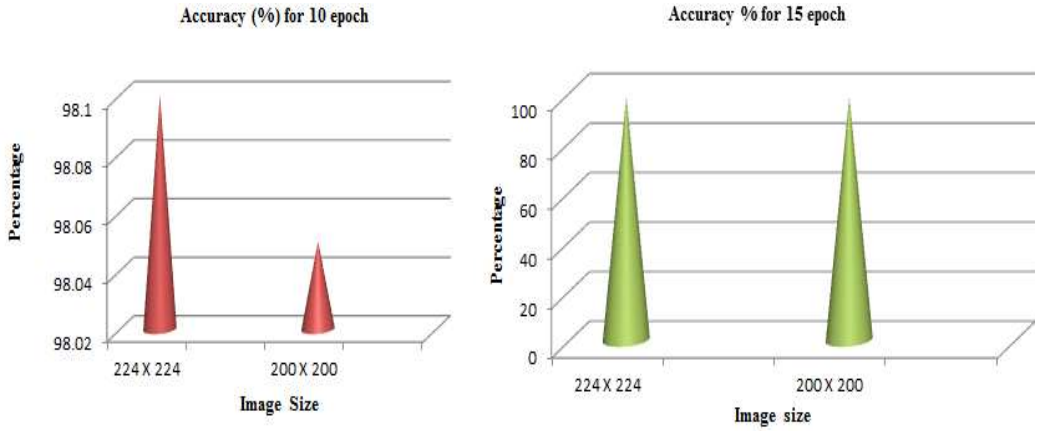



***Figure 5****: Accuracy of the proposed model across epochs and image sizes on the tomato leaf dataset*

Journal of Theoretical and Applied Information Technology
30th April 2026. Vol.104. No.8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

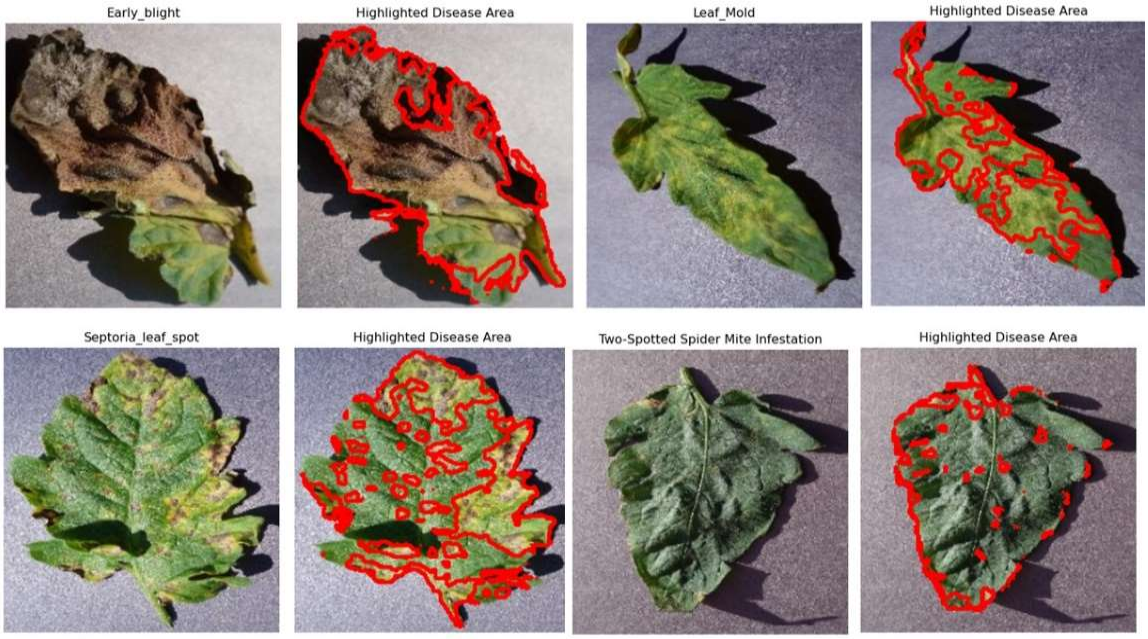



***Figure 6:*** *Some examples of disease-affected areas in tomato leaf images*


***Figure 7****:Dataset classes: (a) bacterial __blight, (b) bacterial blight with leaf _miner, (c) healthy cotton leaf image*

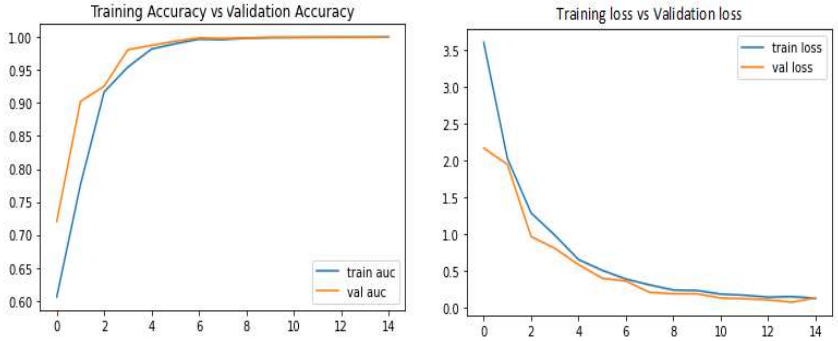



***Figure 8:*** *Training and Validation Accuracy/Loss of Proposed Model on Tomato Leaf Dataset (224×224, 15 Epochs)*

Journal of Theoretical and Applied Information Technology
30th April 2026. Vol.104. No.8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

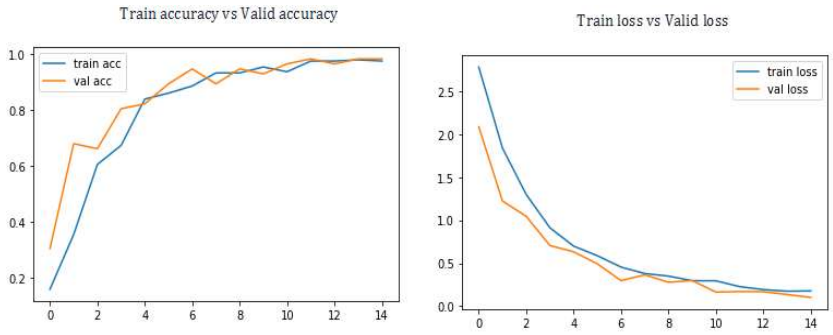



***Figure 9:*** *Training & Validation Accuracy and Loss of Proposed Model (Tomato Leaf, 200×200, 15 Epochs)*

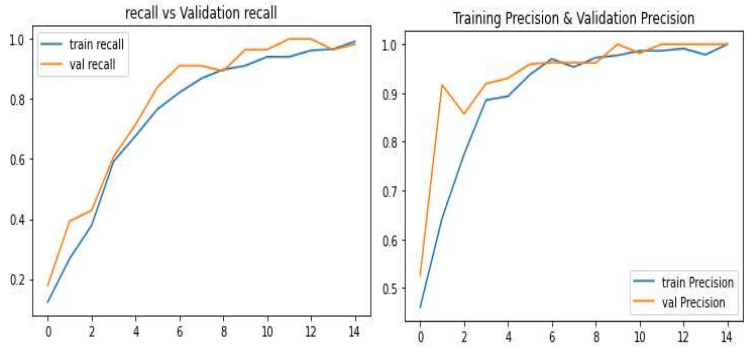



***Figure 10:*** *Training, Validation Recall & Precision of Proposed Model (224×224, Tomato Leaf Dataset)*

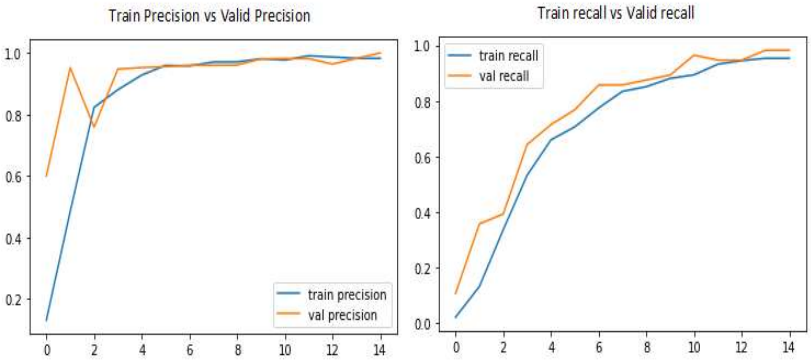



***Figure 11:*** *Training, Validation Recall & Precision of Proposed Model (200×200, Tomato Leaf Dataset)*

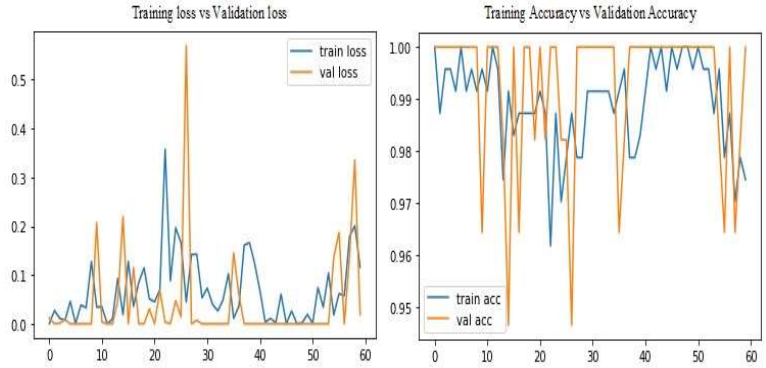



***Figure 12:*** *Training & Validation Accuracy/Loss of Proposed Model on Tomato Leaf Dataset (60 Epochs)*

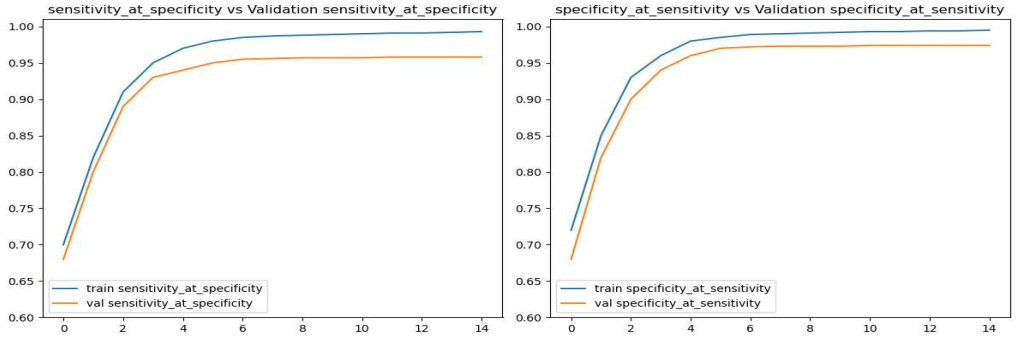



***Figure 13:*** *Training and Validation Sensitivity/Specificity of Proposed Model on Tomato Leaf Dataset*

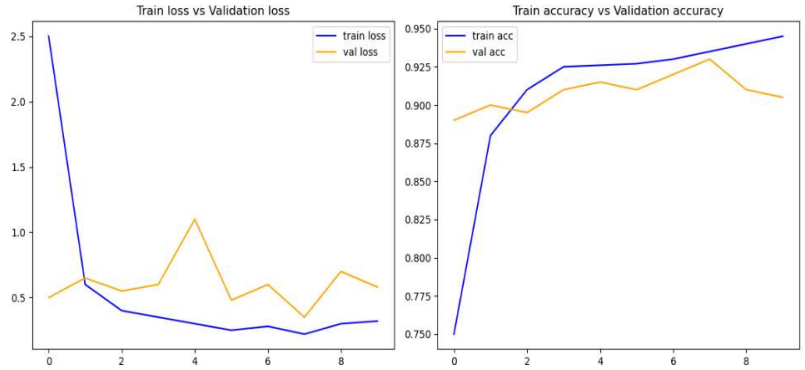



***Figure 14:****Training & Validation Accuracy/Loss of Proposed Model on Cotton Leaf Dataset (10 Epochs)*

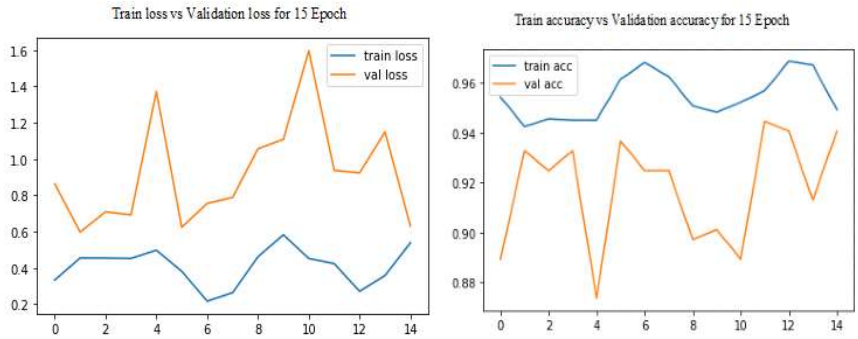



***Figure 15:*** *Training and validation accuracy and loss outcomes of the proposed model on the cotton leaf image dataset over 15 epochs.*

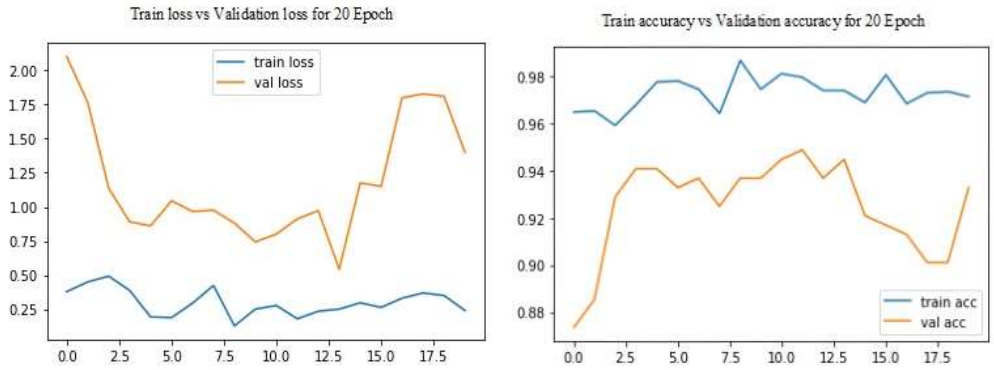



***Figure 16**:Training and validation accuracy and loss outcomes of the proposed model on the cotton image dataset across 20 epochs*

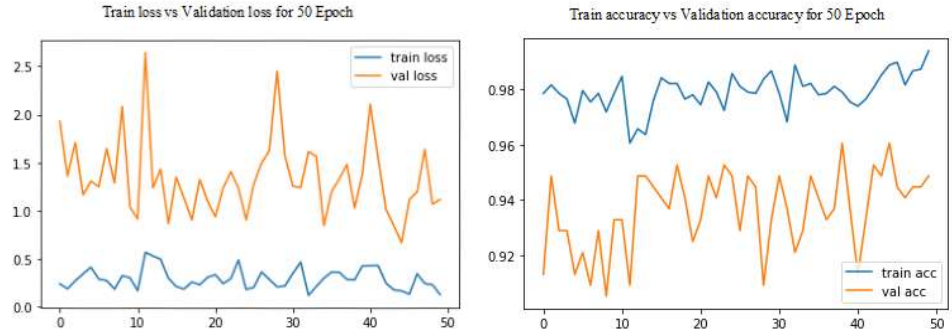
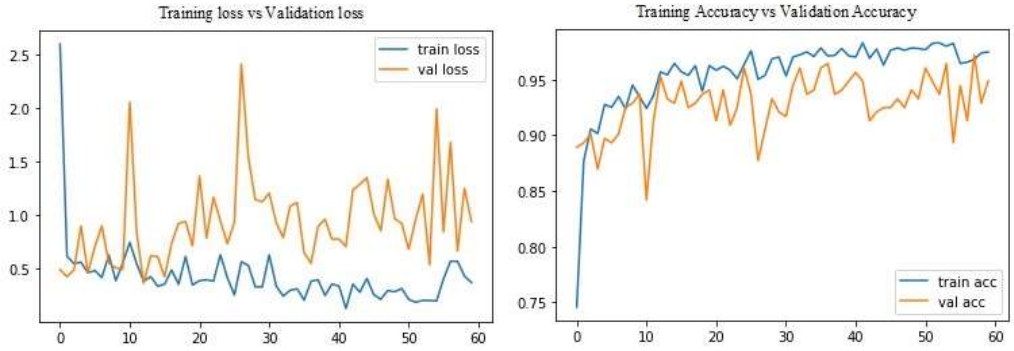


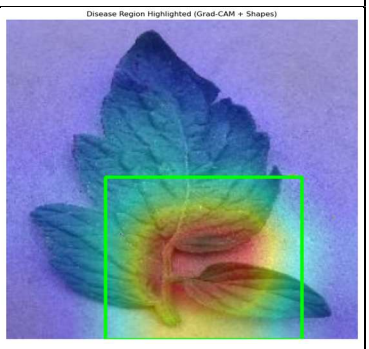
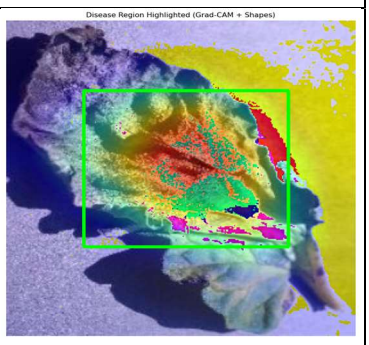
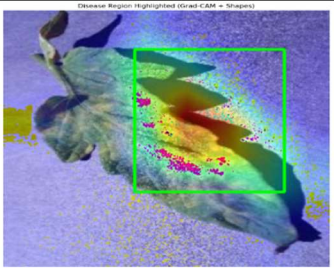
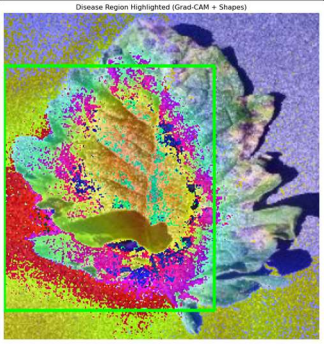

***Figure 17:** Training and validation accuracy and loss outcomes of the proposed model on the cotton image dataset across 50 epochs*



Journal of Theoretical and Applied Information Technology
30th April 2026. Vol.104. No.8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

***Figure 19:*** *Training and validation accuracy and loss outcomes of the proposed model on the cotton leaf image dataset across 60 epochs*

| Class | Input Image | GradCAM Visualization |
|---|---|---|
| Bacterial_spot |  |  |
| Early_blight |  |  |
| Leaf_Mold |  |  |
| Septoria_leaf_spot |  |  |

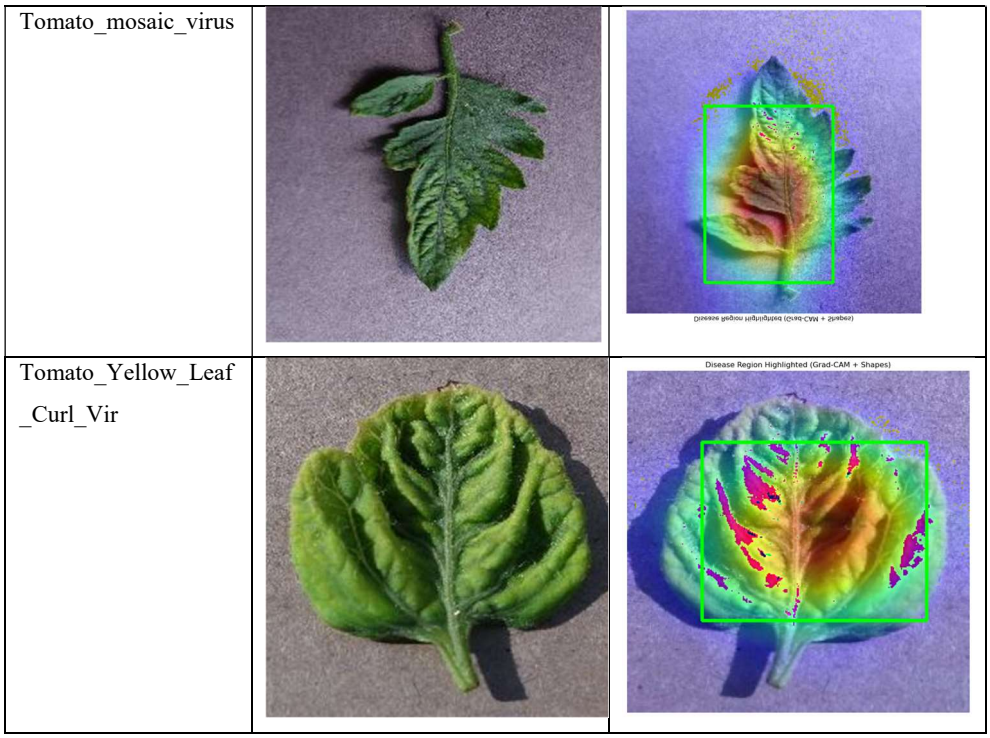



***Figure 20:*** *Visualization of Disease Region Using Grad-CAM*

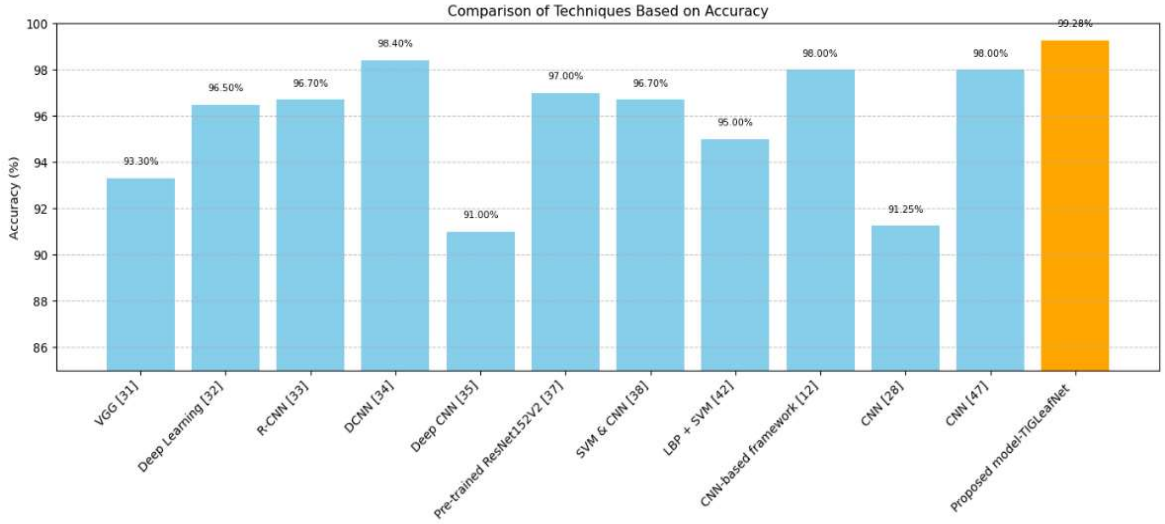



***Figure 21:*** *Comparison of the proposed model for tomato leaf disease with existing methods*

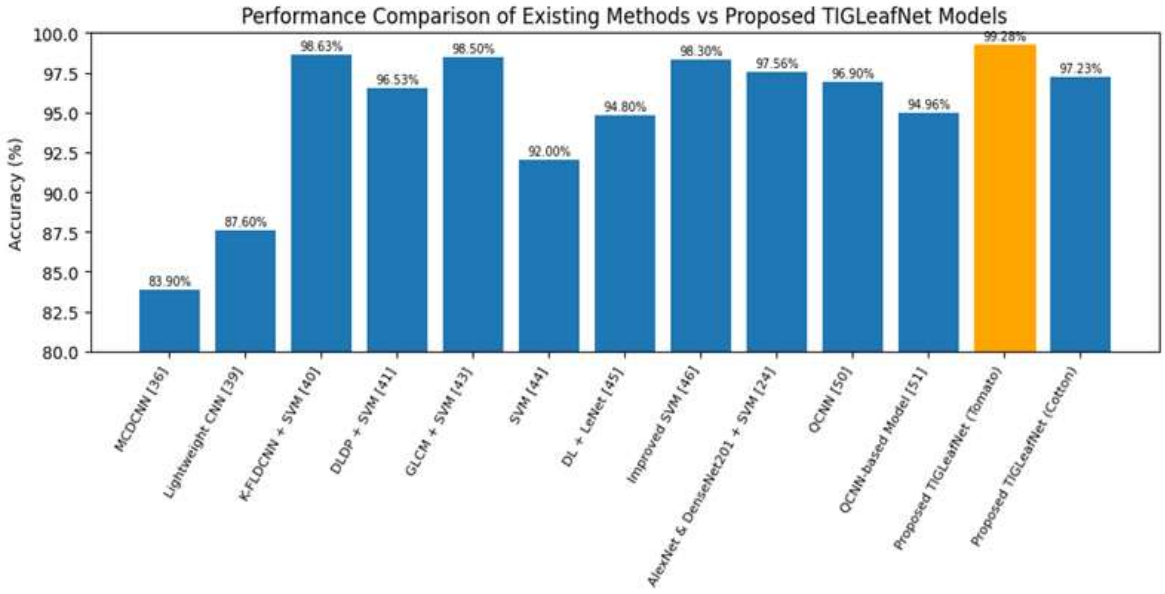



***Figure 22:*** *Accuracy-based comparison of the proposed model with existing methods on the various types of Leaf disease datasets*