ABSTRACT

A common model for classifying images is the convolutional neural network (CNN), which has the benefit of effectively using data correlation information. Despite their remarkable success, classical CNNs may face challenges in achieving further improvements in accuracy, computational efficiency, explainability, and generalization. However, if the specified data dimension or model grows too large, CNN becomes difficult to train effectively with a slowdown processing. In order to address a problem using CNN utilizing quantum computing, Quantum Convolutional Neural Network (QCNN) proposes a novel quantum solution or enhances the functionality of an existing learning model in terms of processing time during training. This paper presents a comparative analysis between classical Convolutional Neural Networks (CNNs) such as DenseNet121, MobileNet, and EfficientNetB0 and a hybrid quantum-classical deep learning model using vision transformer and quantum circuits architecture tailored for image-based tasks, emphasizing the adaptability and versatility of quantum circuits in enhancing feature extraction capabilities and then final accuracy and processing time. The dataset used in this study consists of augmented images of plants, specifically focusing on apple crops. It contains four distinct classes such as Apple__Cedar_apple_rust, Apple_Apple_scab, Apple_Black_rot, Apple__healthy. The analysis was based on training and testing accuracies obtained for five different models: DenseNet121, MobileNet, EfficientNetB0, Hybrid Quantum DenseNet121, and Hybrid Quantum ViT (Vision Transformer). The results indicate that classical models such as DenseNet121, MobileNet, and EfficientNetB0 achieved extremely high training accuracies of 100%, 99.96%, and 90.38% respectively. However, despite these impressive training performances, their testing accuracies remained fixed at 42.86%, which clearly indicates overfitting. In contrast, the Hybrid Quantum DenseNet121 model displayed a more balanced performance with a training accuracy of 70.85% and a testing accuracy of 70.18%. The Hybrid Quantum ViT model achieved a training accuracy of 85.19%, which is relatively high, but it underperformed during testing, achieving only 41.25%. The integration of quantum computing into deep learning architectures opens new frontiers for efficient and accurate disease classification. This project demonstrates that hybrid quantum-classical models are not only theoretically valuable but also practically effective, particularly in precision agriculture and smart farming systems.

Introduction

Agriculture forms the foundation of the global economy, playing a pivotal role in the sustenance of human civilization. It not only provides the food we consume daily and the raw materials essential for various industries but also serves as the primary source of livelihood for a large segment of the world's population, particularly in developing nations. As the global population continues to rise, with projections estimating over 9 billion people by 2050, there is a growing demand for higher agricultural productivity to ensure food security and economic stability.

However, plant diseases pose a significant threat to agricultural productivity. They can cause devastating crop losses, compromise food quality, and negatively impact the agricultural economy. These diseases often spread rapidly and unpredictably, especially in large-scale farming environments, making timely and accurate diagnosis critical. According to reports from the Food and Agriculture Organization (FAO), plant diseases are responsible for up to 40% of global food crop losses annually, underscoring the urgent need for efficient plant health monitoring systems.

Apple Plant Disease:

Apple (Malus domestica) is one of the most popular and widely cultivated fruit crops in the world due to its taste, nutritional value, and economic importance. However, apple plants are highly vulnerable to a variety of diseases that can significantly reduce fruit quality, yield, and tree lifespan. Understanding and managing these diseases is crucial for sustainable apple production and public health.

Causes of Apple Plant Diseases

- Fungal Infections:
  - Apple Scab (caused by Venturia inaequalis)
  - Powdery Mildew (caused by Podosphaera leucotricha)
  - Black Rot (caused by Botryosphaeria obtusa)

Impact on Human Health and Economy

While apple plant diseases mainly affect the plant, their indirect effects on human health and the agricultural economy are significant:

- Reduced Food Quality & Quantity:
  - Infected apples are often smaller, misshapen, or inedible.
  - Leads to reduced availability of healthy fruits rich in fiber and vitamin C.
- Economic Damage:
  - Losses for farmers and fruit processors due to crop failure or rejection in markets.
  - Increased costs for pesticides and disease management.
- Health Risks:
  - Overuse of chemical pesticides can lead to toxic residues on fruit, which may pose long-term health risks to consumers.
  - Fungal infections can sometimes lead to production of mycotoxins, harmful to liver and immune systems if consumed.

Solutions and Prevention Strategies

Effective management of apple diseases involves integrated approaches combining prevention, early detection, and treatment:

1. Cultural Practices:

- Proper pruning to ensure air circulation and sunlight exposure.
- Sanitation: Removing fallen leaves and infected fruits.
- Crop rotation and avoiding planting near old diseased orchards.

2. Chemical Control:

- Timely application of fungicides and bactericides (e.g., copper-based sprays for fire blight).
- Use of organic alternatives when possible.

3. Biological and Eco-Friendly Methods:

- Use of beneficial microbes to combat fungal pathogens.

- Neem oil and compost teas as organic treatments.

4. Disease-Resistant Varieties:

- Cultivating resistant apple cultivars like ‘Liberty’, ‘Enterprise’, and ‘GoldRush’ helps reduce dependency on chemicals.

5. Technological Innovations:

- AI and Machine Learning models for early detection of diseases from leaf images.
- Remote sensing and drones for large-scale orchard monitoring.
- Quantum-classical hybrid models (emerging research area) for precise disease detection and classification.

Challenges in Traditional Diagnosis

Traditionally, the diagnosis of plant diseases relies heavily on manual inspection by agricultural experts. This approach, although reliable to some extent, suffers from several limitations:

- It is time-consuming and labor-intensive, requiring trained personnel to travel and inspect crops manually.
- It is prone to human error and subjectivity, as assessments may vary between individuals.
- In rural and remote areas, the availability of agricultural experts is limited, delaying disease detection and management.

These challenges highlight the need for automated, intelligent, and scalable plant disease detection systems that can operate in real-time and with high accuracy.

Emergence of Image-Based Deep Learning Techniques

With the rise of computer vision and deep learning, particularly Convolutional Neural Networks (CNNs), significant strides have been made in automating the process of plant disease detection through leaf image analysis. CNNs have shown remarkable performance in classifying plant diseases by learning hierarchical features from image data.

However, CNNs have their limitations:

- They are computationally expensive, with a large number of trainable parameters.

- They often require large labeled datasets for training, which may not be readily available in the agricultural domain.
- CNNs are not inherently good at modeling long-range dependencies within image data, which can be critical for nuanced classification tasks.

The Promise of Quantum Computing

- Despite the advancements in classical models like CNNs and ViTs, deploying such models on resource-constrained devices (e.g., edge devices) or in regions with limited computational infrastructure remains a significant challenge.
- This is where quantum computing introduces a transformative potential. Unlike classical computing, quantum computing leverages the principles of superposition, entanglement, and quantum interference to process information in fundamentally different ways. In theory, quantum computers can solve certain problems exponentially faster than their classical counterparts.
- Although we are currently in the Noisy Intermediate-Scale Quantum (NISQ) era — where quantum devices are not yet fault-tolerant or large-scale — hybrid quantum-classical computing presents a practical and forward-looking solution.

Literature Review

In [1], presents a comprehensive analysis of state-of-the-art Convolutional Neural Network (CNN) models for classifying diseases in pumpkin plant leaves. Using a publicly available dataset of 2000 highresolution images, we evaluate the performance of several CNN architectures, including ResNet, DenseNet, and EfficientNet, in recognizing five classes: healthy leaves and four common diseases downy mildew, powdery mildew, mosaic disease, and bacterial leaf spot. ResNet-34, DenseNet-121, and EfficientNet-B7 were identified as top-performing models, each excelling in different classes of leaf diseases. Our analysis revealed DenseNet-121 as the optimal model when considering both accuracy and computational complexity achieving an overall accuracy of 86%.

In [2], a convolutional neural network architecture FL-EfficientNet (Focal loss EfficientNet), which is used for multi-category identification of plant disease images. The experiment uses the public data set new plant diseases dataset (NPDD) and compares it with ResNet50, DenseNet169, and EfficientNet. The experimental results show that the accuracy of FL-EfficientNet in identifying 10 diseases of 5 kinds of crops is 99.72%, which is better than the above comparison network. At the same time, FL-EfficientNet has the fastest convergence speed, and the training time of 15 epochs is 4.7 h.

In [3], presents a comprehensive exploration of utilizing machine learning, specifically Convolutional Neural Networks (CNNs), for accurate and timely plant disease classification. The dataset, encompassing various crops and diseases, forms the foundation for training the model, while preprocessing ensures optimal data quality. The CNN architecture, meticulously designed, progresses through convolutional and pooling layers to extract hierarchical features from input images. The trained model achieves an accuracy of 92.23% on disease classification, showcasing the potential of such technology in agricultural innovation.

In [4], presents a transfer learning-based model for identifying diseases in plant leaves. A CNN classifier based on transfer learning model called DenseNet201 are proposed. An analysis of four deep learning models (VGG16, Inception V3, ResNet152V2, and DenseNet201) done to see which one can detect plant diseases with the greatest degree of accuracy. Web based application developed for plant disease diagnosing from defected leaf image and the proposed model which identify the disease and give the recommended treatment. The used images dataset contains 28310 leaves photos of 3 crops, tomato, potato and pepper divided into 15 different classes, 9 disorders and one healthy class for tomato, 2 disorders and one healthy class for potato and 1 disorder and one healthy for pepper. In our experimental, the results shows that the proposed model achieves the highest training accuracy of 99.44% and validation accuracy of 98.70%.

In [5], introduces a deep dense net slice fragmentation and segmentation feature selection and classification through optimized convolution neural network. Initially the wavelet Filters features are applied to enhance the image through structure normalization model. The slice fragment segmentation is applied to segment the disease covered region by identifying the realistic variation based on spectral histogram feature difference. Then cascaded edges and features are extracted and trained using deep Densenet Convolution Neural Network (DnCNN) to identify the plant disease effectively. The proposed system achieves best result compared to the other existing approaches in terms of precision rate, recall, f-measure and also superior due to the fact that the diseases are identified at an earlier stage.

In [6] An optimized dense convolutional neural network (CNN) architecture (DenseNet) for corn leaf disease recognition and classification is proposed. The current research presents a solution through deep learning so that crop health can be monitored and, it will lead to an increase in the quantity as well as the quality of crop production. The proposed optimized DenseNet model has achieved an accuracy of 98.06%. Besides, it uses significantly lesser parameters as compared to the various existing CNN such as EfficientNet, VGG19Net, NASNet, and Xception Net. The performance of the optimized DenseNet model has been contrasted with the current CNN architectures by considering two (time and accuracy) quality measures. This study indicates that the performance of the optimized DenseNet model is close to that of the established CNN architectures with far fewer parameters and computation time.

In[7] EfficientNet deep learning architecture was proposed in plant leaf disease classification and the performance of this model was compared with other state-of-the-art deep learning models. The PlantVillage dataset was used to train models. All the models were trained with original and augmented datasets having 55,448 and 61,486 images, respectively. EfficientNet architecture and other deep learning models were trained using transfer learning approach. In the transfer learning, all layers of the models were set to be trainable. The results obtained in the test dataset showed that B5 and B4 models of EfficientNet architecture achieved the highest values compared to other deep learning models in original and augmented datasets with 99.91% and 99.97% respectively for accuracy and 98.42% and 99.39% respectively for precision.

In [8] developed a novel CNN model that is suitable for small-scale farmers. The numerical outcomes indicate that the proposed model surpassed the state-of-the-art models by achieving an average accuracy of 96.86%. The proposed model utilized comparatively limited computational resources as analyzed through floating-point operations (FLOPs), number of parameters, computation time, and model's size. Furthermore, a statistical approach was proposed to analyze a model while collectively accounting for its performance and computational complexity. It is observed from the results that the proposed model outperformed the state-of-the-art techniques in terms of both average recognition accuracy and computational complexity.

[9] proposes a method of fine-tuning model parameters based on transfer learning EfficientNet, which can improve the accuracy and speed of network recognition for a small sample of maize disease dataset. First of all, perform data cleaning and data augmentation on the dataset to obtain richer image data; then, transfer the pre-trained model obtained by EfficientNet training on ImageNet to this model method; finally, the last layer of EfficientNet classifier replace with 8 classes of softmax classifier, and train the entire network to obtain a training model for maize disease prediction. In order to verify the robustness and accuracy of the method proposed in this paper, test verification was carried out in the test dataset with VGG-16, Inception-v3 and Resnet-50, respectively. The experimental results show that the training speed of the network model proposed in this paper has been significantly improved, and its recognition accuracy is far better than other networks with a maximum of 98.52%, which can realize agricultural production applications.

In [10] presents a hybrid approach combining MobileNetV2 with a compact CNN architecture, integrated with Local Interpretable Model-agnostic Explanations (LIME) for enhanced interpretability. The model achieved 95% accuracy and 94% precision in disease classification, demonstrating a balance between high accuracy and computational efficiency.

In [11] introduces RTR_Lite_MobileNetV2, an enhanced version of MobileNetV2 optimized with attention mechanisms for deployment on resource-constrained devices. The model achieved high accuracies across multiple datasets, such as 99.92% on Plant Disease and 97.11% on PaddyDoctor, demonstrating its efficiency and suitability for real-time plant disease diagnosis.

In [12] Quantum Computing is a technology, which promises to overcome the drawbacks of conventional CMOS technology for high density and high performance applications. Its potential to revolutionize today's computing world is attracting more and more researchers towards this field. However, due to the involvement of quantum properties, many beginners find it difficult to follow the field. Therefore, in this research note an effort has been made to introduce the various aspects of quantum computing to researchers, quantum engineers and scientists. The historical background and basic concepts necessary to understand quantum computation and information processing have been introduced in a lucid manner. Various physical implementations and potential application areas of quantum computation have also been discussed in this paper. Recent developments in each realization, in the context of the DiVincenzo criteria, including ion traps based quantum computing, superconducting quantum computing, nuclear magnetic resonance (NMR) quantum computing, spintronics and semiconductor based quantum computing have been discussed.

In [13] using machine learning for image classification is very common. However, due to the increasing demand for data processing and fast computing, the idea of enhancing machine learning with quantum computing has been proposed, known as quantum machine learning (QML). Quantum machine learning has the advantages of higher efficiency and accuracy. Quantum computing uses quantum bits (qubits) for data storage and computing, where a qubit can represent quantum states $|0\rangle$ and $|1\rangle$ simultaneously, enabling the processing of information for two states simultaneously, which is unparalleled in classical computing. Moreover, quantum machine learning can handle more complex data and process data faster. In classical machine learning, the processing of large-scale data and complex problems often faces problems of high computational complexity and low algorithm efficiency. Quantum computing can handle multiple computing tasks simultaneously, achieving faster computing. Therefore, in some scenarios that require efficient computing, quantum machine learning may be the best choice. In this study, we simulated quantum circuits using Qiskit and built a hybrid quantum-classical neural network model using VQNet to classify MNIST handwritten digits and CIFAR-10 datasets. The experiments showed that quantum machine learning has the advantages of efficiency, accuracy, and security over classical machine learning, which may be an improvement over classical machine learning. This research proposes a machine learning algorithm based on quantum computing, which promotes the development of quantum computing and quantum technology. At the same time, it provides a new solution and idea for image classification, enabling people to pursue faster and more accurate quantum machine learning instead of being limited to classical machine learning.

In [14] introduces a hybrid quantum-classical convolutional neural network (QC-CNN) that applies quantum computing to effectively extract high-level critical features from EO data for classification purposes. Besides that, the adoption of the amplitude encoding technique reduces the required quantum bit resources. The complexity analysis indicates that the proposed model can accelerate the convolutional operation in comparison with its classical counterpart. The model's performance is evaluated with different EO benchmarks, including Overhead-MNIST, So2Sat LCZ42, PatternNet, RSI-CB256, and NaSC-TG2, through the TensorFlow Quantum platform, and it can achieve better performance than its classical counterpart and have higher generalizability, which verifies the validity of the QC-CNN model on EO data classification tasks.

In [15] is devoted to the development and application of quantum methods in the field of diagnostics of infectious diseases of wheat. Quantum image processing algorithms have been developed to improve quality, filter noise, and high- light key features in photographs of samples. Quantum filters and contrast enhancement techniques were used. The application of Quantum Fourier Transform (QFT) to image processing is an innovative technique based on the principles of quantum computing. QFT can be used to analyse the frequency characteristics of an image, which is useful in the context of signal processing and extracting key features. Quantum representation of an image. The image is converted to a quantum format, where each pixel is represented as a qubit state. This representation makes it possible to process large amounts of data in parallel. Each qubit is initialized according to the brightness of the corresponding pixel in the image. Thus, the state of the system is a quantum vector encodeing the intensity of the pixels. Quantum circuit is built from basic quantum operations such as Hadamard gates and controlled phase gates to implement the quantum Fourier transform.

Literature Review Table

S. No	Title	Author	Year	Model	Findings
1.	Automated disease diagnosis in pumpkin plants using advanced cnn models	Aymane khaldi, el mostafa kalmoun	2024	Resnet-34, densenet-121, and efficientnet-b7	Densenet-121 as the optimal model achieving an overall accuracy of 86%
2.	Research on plant disease identification based on cnn	Xuewei sun, Guohou li, Peixin qu, Xiwang xie, Xipeng pan, Weidong zhang	2022	resnet50, densenet169, and efficientnet.	Accuracy of fl-efficientnet in identifying 10 diseases of 5 kinds of crops is 99.72%
3.	Leaf disease detection by convolutional neural network (cnn)	Shutuo guo	2023	Cnn model	The trained model achieves an accuracy of 92.23% on disease classification
4.	Densenet based model for plant diseases diagnosis	Mahmoud bakr, Sayed abdel-gaber mona nasr Maryam hazman	2022	Vgg16, inception v3, resnet152v2, and densenet201	Proposed model (densenet201) achieves the highest training accuracy of 99.44% and validation accuracy of 98.70%.
5.	Plant leaf disease prediction using deep dense net slice fragmentation and segmentation feature selection using convolution neural network	S. Jana, s. D. Thilagavathy, s. T. Shenbagavalli, g. Srividhya, v. S. Gowtham prasad, r. Hemavathy	2023	Deep densenet convolution neural network (dncnn)	Proposed system achieves best result compared to the other existing approaches in terms of precision rate, recall, f-measure.

6.	An optimized dense convolutional neural network model for disease recognition and classification in corn leaf	Abdul waheed, Muskan goyal, Deepak gupta, Ashish khanna, Hari mohan pandey	2020	Densenet	Densenet model has achieved an accuracy of 98.06%.
7.	Plant leaf disease classification using efficientnet deep learning model	Umit atila, Murat ucar, Kemal akyol, Emine uçar	2021	Efficientnet	Efficientnet architecture achieved in original and augmented datasets with 99.91% and 99.97% respectively for accuracy and 98.42% and 99.39% respectively for precision
8.	Automatic plant disease detection using computationally efficient convolutional neural network	Muhammad rizwan, samina bibi, sana ul haq, muhammad asif, tariqullah jan, mohammad haseeb zafar	2024	Novel cnn model	Achieving an average accuracy of 96.86%.
9.	Efficientnet-based recognition of maize diseases by leaf image classification	Liu jiangchuan and wang mantao	2020	Efficientnet	Accuracy of 98.52%.
10.	Plant disease detection using hybrid mobilenetv2-compact cnn architecture with lime integration	Ramakrishna badiguntla, Y. Narasimha rao, Srees boppana	2025	Mobilenetv2	The model achieved 95% accuracy and 94% precision in disease classification
11.	Rtr_lite_mobilenetv2: a lightweight and efficient model for plant disease	Sangeeta duhan, preeti gulia, nasib singh gill, ekta narwal	2025	Rtr_lite_ Mobilenetv2	99.92% accuracy achieved on plant disease and 97.11% on paddy disease.

	detection and classification				
12.	Quantum computing: fundamentals, implementations and applications	Hilal ahmad bhat, farooq ahmad khanday, brajesh kumar kaushik, faisal bashir 1, and khurshed ahmad shah	2022	Quantum Model	Quantum computing, superconducting quantum computing, nuclear magnetic resonance (NMR) quantum computing, spintronics and semiconductor based quantum computing have been discussed.
13.	Image classification based on quantum machine learning	Haocheng xiong; xinyuan duan; yue yu; junhan zhang; hao yin	2023	Quantum circuits and hybrid quantum neural model	quantum machine learning has the advantages of efficiency, accuracy, and security over classical machine learning
14.	Hybrid quantum-classical convolutional neural network model for image classification	Fan fan, yilei shi, Tobias guggemos, Xiao xiang zhu	2023	hybrid quantum-classical convolutional neural network (QC-CNN)	it achieves better performance than its classical counterpart and have higher generalizability
15.	Application of quantum computing in image processing for recognition of infectious diseses of wheat	D t mukhamedieva and r a sobirov	2024	Quantum fourier transform (qft), quantum model	Analysis of amplitudes and phases allows you to identify key features of the image.

Scope of Work

Automate Plant Disease Detection Using Deep Learning Models

The system leverages advanced deep learning architectures—DenseNet121, MobileNet, and EfficientNet—each pre-trained and fine-tuned for plant leaf image classification. These models detect and categorize various plant diseases based on visual patterns in leaf images.

- Preprocessing of leaf images (resizing, normalization, etc.).
- Training and inference using CNN-based models optimized for accuracy and efficiency.
- Prediction of disease type with confidence scores.
- Handling multiple disease categories across plant types.

This automation enables fast, consistent, and accurate plant disease recognition, reducing reliance on manual agricultural inspection.

Reduce Time and Effort for Farmers and Agronomists

By automating disease identification, the system significantly lowers the effort and expertise required by farmers to diagnose plant health issues. Early detection helps prevent the spread of diseases and supports timely intervention.

- Instant disease classification upon image upload.
- Summary output displaying predicted disease and probability.
- Exportable disease reports for consultation or expert review.

This reduces diagnostic time, increases crop yield potential, and supports sustainable agriculture.

Enable Lightweight and Efficient On-Device Diagnosis

MobileNet and EfficientNet models are optimized for low-resource environments. Their lightweight nature makes them suitable for deployment on mobile devices, ensuring accessibility for field use.

- Real-time classification on edge devices or smartphones.
- Minimal computational resources required for inference.

- Ideal for remote and resource-limited settings.

This makes the technology highly usable in rural farming areas without constant internet or high-end devices.

Integrate Hybrid Quantum-Classical Model for Future-Ready Solutions

To explore cutting-edge innovation, the system also incorporates a hybrid quantum-classical model using a Vision Transformer (ViT) and quantum layers. This experimental model aims to assess the benefits of quantum computing in agricultural AI.

- Uses quantum circuits to enhance decision-making layers.
- Integration with PennyLane and PyTorch for hybrid execution.
- Evaluation of model accuracy and complexity trade-offs.

This step positions the project at the intersection of machine learning and quantum computing, offering forward-looking potential.

Provide Explainable Results and Comparative Insights

The system emphasizes interpretability and transparency in its predictions. Model outputs are designed to support decision-making with visual and comparative analysis.

- Visual representation of classification confidence.
- Comparative tables showing performance of each model.
- Analysis highlighting strengths and limitations per architecture.

These insights build user trust and guide the selection of the best-fit model for specific deployment scenarios.

Existing Model and Need for a system

Existing Models

Several deep learning-based models have been previously proposed for plant disease detection, primarily using convolutional neural networks (CNNs). Key existing approaches include:

1. **VGGNet and AlexNet-Based Models**
   - Early research adopted these deep CNN architectures to classify plant diseases.
   - These models demonstrated good accuracy but required significant computational resources and had limitations in terms of depth and overfitting on small datasets.
2. **ResNet and Inception Models**
   - Later models introduced residual learning (ResNet) and inception modules to improve depth and feature extraction without drastically increasing complexity.
   - These models improved classification performance but still posed challenges in deployment on mobile or embedded devices.
3. **Custom Lightweight CNNs**
   - Some researchers proposed custom CNN architectures tailored to specific datasets or plant types.
   - While faster, these often compromised on accuracy or lacked generalizability across different species.
4. **Transfer Learning with Pre-Trained Models**
   - Approaches using pre-trained models such as DenseNet, EfficientNet, and MobileNet have gained popularity for their balance of speed and accuracy.
   - They benefit from large-scale pretraining on ImageNet and can be fine-tuned effectively on agricultural datasets.

Despite these advancements, most existing systems are classical and do not explore integration with modern technologies like quantum computing, nor do they offer cross-comparison across multiple architectures under the same conditions.

Need for the System

There is a pressing need for a robust, intelligent, and scalable system for plant disease detection due to the following reasons:

1. **Early Detection and Prevention**
   - Timely identification of diseases is crucial for preventing crop loss, especially in regions dependent on agriculture for livelihood.
2. **Lack of Expert Availability**
   - In many rural and under-resourced areas, access to trained agronomists or plant pathologists is limited, making automated systems highly valuable.
3. **Need for Model Comparisons and Benchmarking**
   - Most prior works focus on a single model, but there is a lack of side-by-side performance comparison among modern CNNs and hybrid models to identify optimal solutions for different needs.
4. **Exploring Quantum Computing in Agriculture**
   - The potential of quantum-enhanced models is still underexplored in agricultural AI. A hybrid quantum-classical approach opens up new research and performance possibilities.
5. **Improved Accuracy, Speed, and Interpretability**
   - Farmers and stakeholders need systems that are not only accurate but also fast and interpretable for practical decision-making.

Operating Environment – Hardware and Software

This section outlines the required hardware and software environment for effective development, testing, and deployment of the Plant Disease Detection System using deep learning and hybrid quantum-classical models. The project involves computationally demanding tasks such as image classification and quantum simulation, necessitating a robust infrastructure.

Hardware Requirements

To ensure smooth operation, scalability, and efficient inference, the following system configurations are recommended:

Minimum Configuration

- **RAM**: 8 GB
- **GPU**: At least 2 GB VRAM (e.g., NVIDIA GTX 1650 or equivalent)
- **Processor**: Intel Core i5 / apple M1/ AMD Ryzen 5 or equivalent
- **Storage**: 10 GB free disk space for dataset, models, and outputs
- **Display**: 1080p or higher (to visualize disease-affected regions clearly)

Recommended Configuration (For Faster Inference and Quantum Simulation)

- **RAM**: 32 GB or higher
- **GPU**: NVIDIA RTX 3090 / A100 / integrated GPU
- **Processor**: Intel Core i9 / AMD Ryzen 9 or better
- **Storage**: SSD with 500+ GB for fast data processing
- **Cooling System**: Effective GPU/CPU cooling for long-duration training and inference tasks

Software Requirements

The project utilizes modern open-source tools and platforms for computer vision, deep learning, and quantum simulation tasks.

Programming Language

- **Python 3.8+**

  Python is a high-level, easy-to-learn programming language widely used in data science, machine learning, and quantum computing.

In our project, Python is used for:

- Writing code for deep learning and quantum-classical models.
- Using libraries like PyTorch, PennyLane, NumPy, and Pandas.
- Data preprocessing, training, and evaluation of models.
- Visualization of results using Matplotlib and Seaborn.
- Python's simplicity, flexibility, and rich ecosystem of libraries make it ideal for developing advanced AI and quantum models.

Libraries & Frameworks

- **PyTorch**: PyTorch is a flexible, open-source deep learning framework used in this project for building and training both classical and hybrid quantum-classical models.
  - Dynamic Graphs: Enables easy experimentation with complex architectures like DenseNet, ViT, and quantum layers.
  - Pre-trained Models: Supports transfer learning using models like DenseNet121, EfficientNetB0, and MobileNet.
  - Quantum Integration: Seamlessly integrates with PennyLane to create hybrid models combining classical CNNs and quantum circuits.
  - GPU Acceleration: Speeds up training of deep models.
  - Custom Data Handling: Efficiently loads and processes plant/X-ray images using DataLoader.
  - PyTorch forms the core of your model pipeline—from data preprocessing to training and inference in both classical and quantum learning workflows.

- **Torchvision**: Torchvision is a PyTorch library that provides tools for computer vision tasks. In our project, it's used for:
  - Datasets: Easily load popular image datasets like ImageNet, CIFAR, or custom datasets.
  - Transforms: Apply preprocessing like resizing, normalization, and data augmentation (e.g., flip, rotate) to input images.
  - Models: Access pretrained CNN models like DenseNet, MobileNet, and EfficientNet for transfer learning.
  - It helps you prepare image data and use deep learning models efficiently in tasks like plant disease classification.

- **PennyLane**: PennyLane is an open-source Python library for hybrid quantum-classical machine learning, developed by Xanadu.

  In this project,

  - PennyLane is used to integrate quantum circuits into classical deep learning models built with PyTorch.
  - Quantum Circuit Design: Allows creation of parameterized quantum circuits (e.g., using qubits, rotations, entanglement).
  - Hybrid Learning: Supports combining classical layers (CNN, ViT) with quantum layers using TorchLayer.
  - Backpropagation Support: Enables automatic differentiation through both classical and quantum parts of the model.
  - Multiple Quantum Backends: Can run on simulators or real quantum hardware (e.g., IBMQ, Rigetti, Xanadu).
  - In our project, PennyLane is key to creating and training the quantum layer in the hybrid quantum-classical model for tasks like plant disease image classification.

- **PennyLane-Lightning:** PennyLane-Lightning is a high-performance version of PennyLane designed for quantum computing simulations. It is an optimized backend for running quantum circuits on classical hardware, offering faster execution and reduced memory consumption compared to the default PennyLane simulator.

  Key Features:

  - Faster Execution: PennyLane-Lightning is optimized for speed, allowing more efficient simulation of quantum circuits.
  - Low Resource Usage: Consumes less memory and computational resources, making it ideal for large-scale quantum simulations.
  - Compatibility: It works seamlessly with PennyLane's full ecosystem, allowing you to build hybrid quantum-classical models and integrate with frameworks like PyTorch.
  - In our project, PennyLane-Lightning can be used to simulate quantum circuits efficiently while training your hybrid quantum-classical models without relying on cloud-based quantum hardware, speeding up experiments.

- **Matplotlib**: Matplotlib is a widely-used Python library for creating visualizations such as plots, graphs, and charts.

  In this project, it is used to:

- Visualize Training Progress: Plot training and validation accuracy/loss over epochs.
- Display Sample Images: Show input images (like plant leaves or X-rays) with predictions.
- Compare Model Results: Plot bar charts or confusion matrices to compare model performance.
- Matplotlib helps you analyze and present your model's behavior and results clearly and effectively.

- **Numpy**: NumPy (Numerical Python) is a fundamental Python library for numerical computing. In this project, NumPy is used to:
- Handle Arrays: Efficiently manage and manipulate image data and model outputs as arrays.
- Preprocess Data: Perform operations like reshaping, normalization, and mathematical computations.
- Support ML Models: Many libraries (like PyTorch, Matplotlib) internally use NumPy arrays for data handling.
- NumPy makes data manipulation faster, simpler, and more efficient, especially when working with large datasets like images.

- **Pandas**: Pandas is a powerful Python library for data manipulation and analysis, particularly useful for working with structured data like CSV files, spreadsheets, or tabular data.
  In this project, Pandas is used to:
- Load and Handle Datasets: Import datasets (e.g., labeled images or metadata) for training and evaluation.
- Data Preprocessing: Clean and preprocess data, such as handling missing values, encoding labels, and organizing image file paths.
- Analysis: Analyze the performance of your models using statistical methods and summary metrics.
- Pandas enables us to easily manage data workflows, preparing it for further machine learning or quantum tasks.

- **Seaborn**: Seaborn is a Python data visualization library based on Matplotlib, designed for making statistical graphics with a high-level interface.
  In this project, Seaborn is used to:
- Create Beautiful Visuals: Generate detailed and attractive plots like heatmaps, correlation matrices, and distribution plots for model performance analysis.
- Visualize Data: Easily visualize relationships between features in your datasets (e.g., class distributions, pixel intensities in images).

- Plot Evaluation Metrics: Display confusion matrices or ROC curves to evaluate the performance of your models.
- Seaborn makes it easy to generate insightful, publication-quality plots with less code than Matplotlib.

- **TensorFlow**: TensorFlow is an open-source deep learning framework developed by Google. It provides a comprehensive ecosystem for building, training, and deploying machine learning models.

  In our project, TensorFlow can be used for:
  - Building Models: Create complex neural networks like CNNs or hybrid quantum-classical models.
  - Training and Optimization: Utilize high-level APIs (like Keras) for model training, hyperparameter tuning, and optimization.
  - Deployment: TensorFlow facilitates easy deployment of models across platforms (e.g., mobile, cloud) via TensorFlow Lite or TensorFlow.js.
  - While PyTorch is the main framework in your project, TensorFlow may also be used for model comparisons or different deployment scenarios.

- **Keras**: Keras is an open-source high-level neural networks API, running on top of frameworks like TensorFlow. It simplifies the process of building and training deep learning models with a user-friendly interface.

  In our project, Keras is used for:
  - Model Construction: Define and build neural network architectures easily using Sequential and Functional APIs.
  - Training: Simplify the training process with built-in functions for compiling, fitting, and evaluating models.
  - Integration with TensorFlow: Since Keras is integrated with TensorFlow, it provides an efficient, high-performance backend for large-scale training tasks.
  - Keras accelerates the development of deep learning models, especially for rapid prototyping and experimentation.

- **Qiskit**: Qiskit is an open-source framework developed by IBM for quantum computing. It allows us to design, simulate, and run quantum algorithms, providing tools for both quantum circuit design and quantum hardware access.

- Quantum Circuits: Build and manipulate quantum circuits with basic gates (e.g., Hadamard, CNOT) and operations.
- Simulation: Test quantum circuits on classical simulators before running on real hardware.
- Quantum Algorithms: Implement algorithms like Grover's and Shor's, useful for complex computational tasks.
- Hybrid Quantum-Classical: Integrate quantum circuits with classical machine learning models (like CNNs or neural networks) for hybrid models.
- In our project, Qiskit helps design quantum layers, run algorithms, and integrate quantum computing with classical machine learning for tasks like plant disease detection.

Development Environment

- **Google Colab - Google Colaboratory (Colab)** is a free, cloud-based Jupyter notebook environment provided by Google. It allows you to write and execute Python code in your browser with no setup required, offering free access to GPUs and TPUs.

  In our project, Google Colab is used for:
  - Writing and running Python code for training and testing both classical and quantum machine learning models.
  - Leveraging GPU acceleration to speed up model training and evaluation processes.
  - Seamless integration with Google Drive for saving datasets, notebooks, and results.
  - Running quantum simulations using libraries like PennyLane and Qiskit without needing local installation.
  - Collaborating in real-time with peers by sharing and editing notebooks simultaneously.
  - Google Colab supports efficient experimentation, rapid prototyping, and collaborative research in machine learning and quantum computing projects.

- **VS Code**: VS Code is a lightweight and powerful source code editor developed by Microsoft. It supports many programming languages and tools, making it ideal for development.

  In our project, VS Code is used for:
  - Writing and editing Python code for classical and quantum models.
  - Running and debugging scripts efficiently with built-in terminal and debugger.
  - Using extensions for Python, Jupyter, Git, and Qiskit/PennyLane for smooth development.
  - Project organization with folder structure, code navigation, and version control.
  - VS Code helps in streamlining coding, debugging, and managing your machine learning and quantum projects effectively.

1. Model Architecture:

1. DenseNet121 (Densely Connected Convolutional Networks)

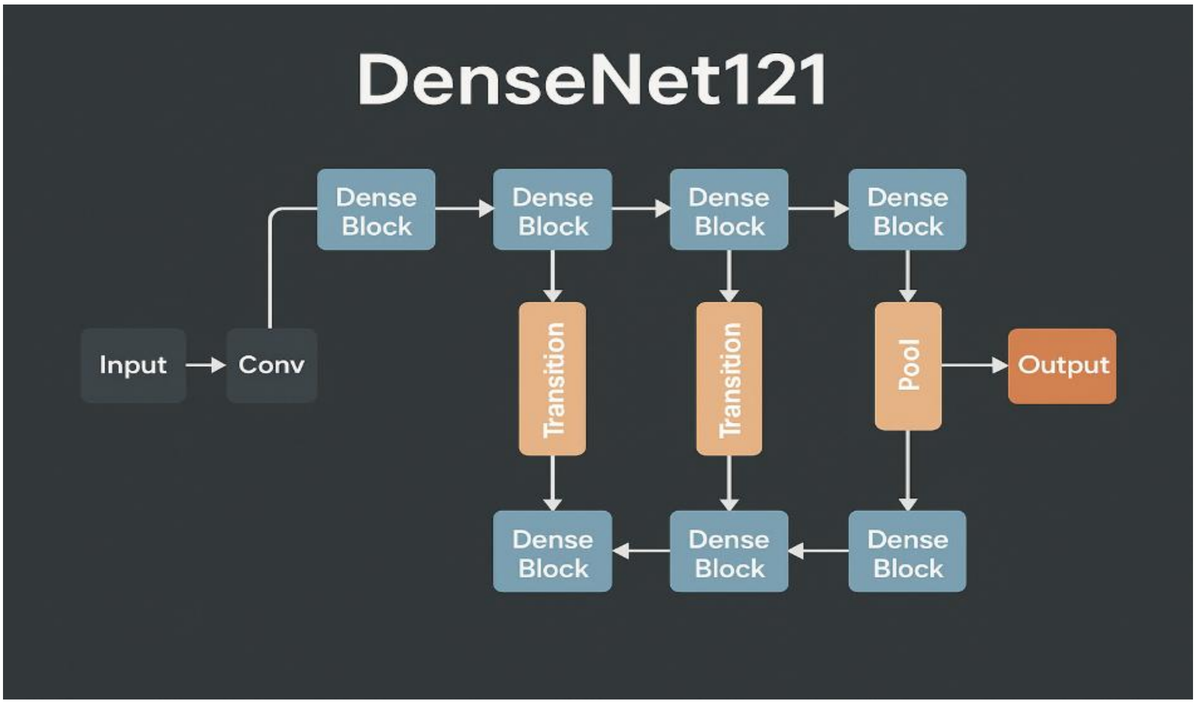
Overview: DenseNet121 is a convolutional neural network with 121 layers, known for its dense connectivity pattern where each layer receives input from all preceding layers.

Key Features:

- **Dense Blocks**: Within each dense block, the feature maps of all previous layers are concatenated and used as input for the next layer.
- **Fewer Parameters**: Despite its depth, it uses fewer parameters due to feature reuse, which also mitigates the vanishing gradient problem.
- **Better Feature Propagation**: Because of its connectivity, features are directly shared across layers, improving learning and accuracy.

Use in Plant Disease Detection:

- Strong at extracting fine-grained features (e.g., leaf lesions).
- Pre-tained on ImageNet and performs well in transfer learning scenarios.



2. MobileNet

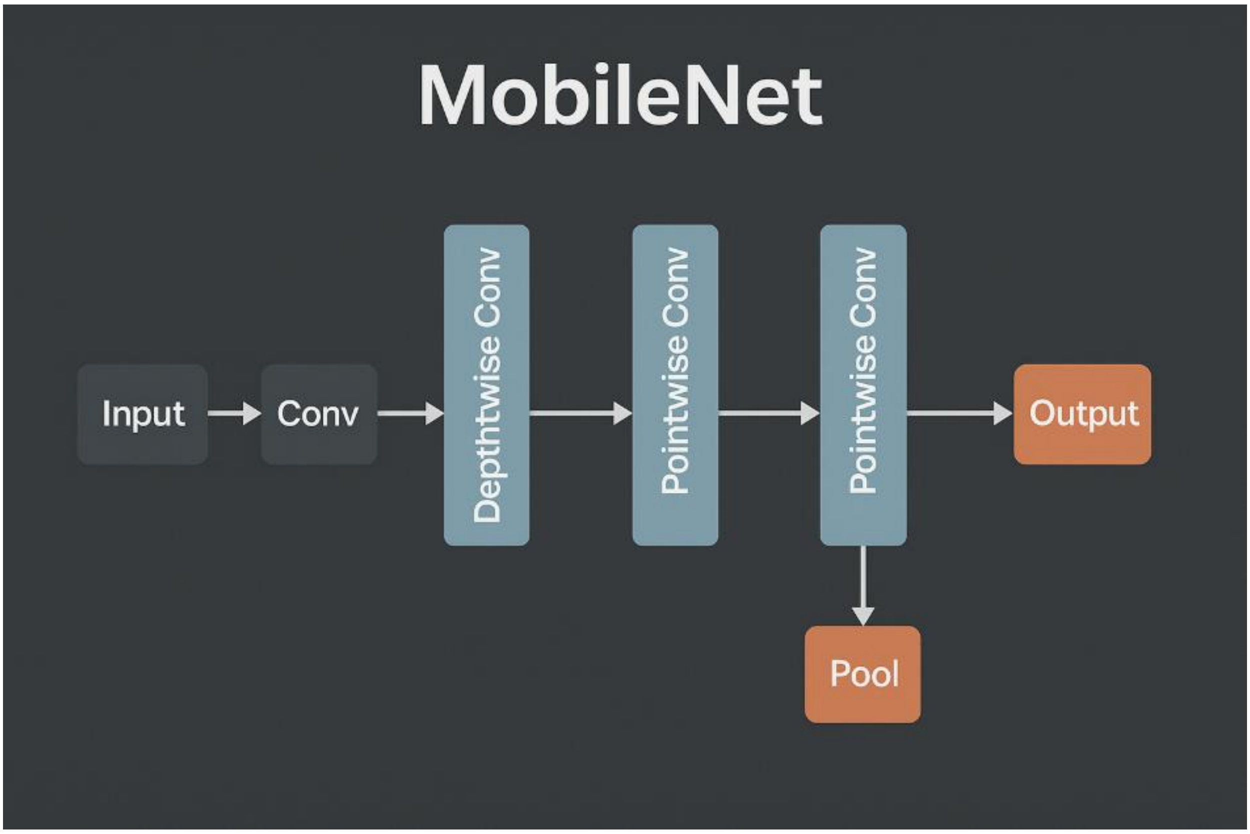
Overview: MobileNet is a lightweight CNN architecture designed for mobile and embedded vision applications, focusing on speed and efficiency.

Key Features:

- **Depthwise Separable Convolutions**: Splits standard convolutions into depthwise and pointwise operations, greatly reducing computation.
- **Low Latency & Memory Footprint**: Ideal for real-time applications or deployment on edge devices.
- **MobileNetV1/V2/V3**: Different versions offer trade-offs in accuracy vs efficiency.

Use in Project:

- Good for on-device disease detection, like in mobile apps for farmers or field doctors.
- Can be integrated with quantum layers in hybrid models due to its small and efficient nature.



3. EfficientNetB0

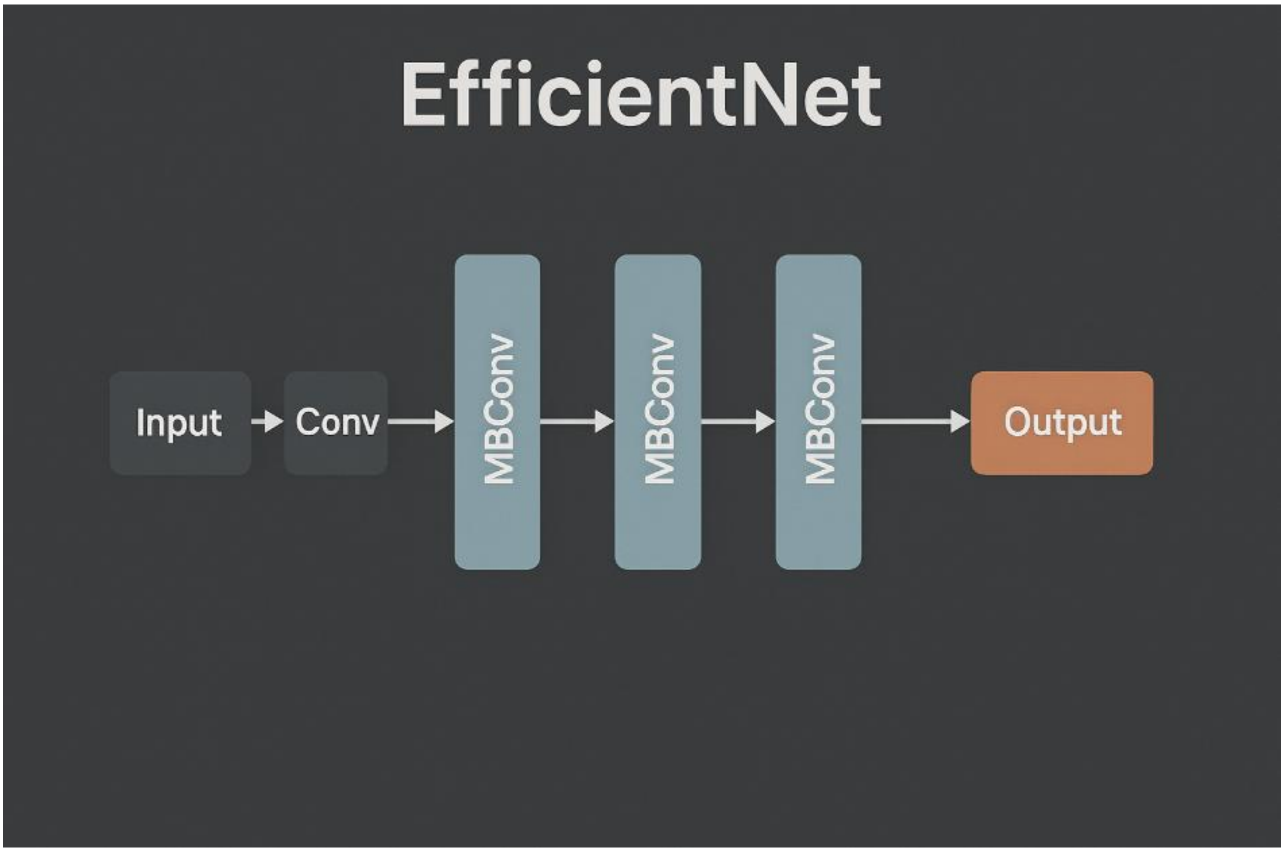
Overview: EfficientNetB0 is the baseline model of the EfficientNet family, which scales model dimensions (depth, width, resolution) using a **compound scaling method**.

Key Features:

- **Compound Scaling**: Uniformly scales depth, width, and resolution with a set of fixed scaling coefficients.
- **Efficient and Accurate**: Provides better accuracy with fewer parameters and less FLOPS compared to older models.
- **MBConv Blocks**: Inverted residual blocks with squeeze-and-excitation layers.

Use in Project:

- Excellent for fine-grained image classification tasks (distinguishing between plant diseases).
- Offers a good balance between performance and computational cost.



4. Hybrid Quantum-DenseNet121

Overview:

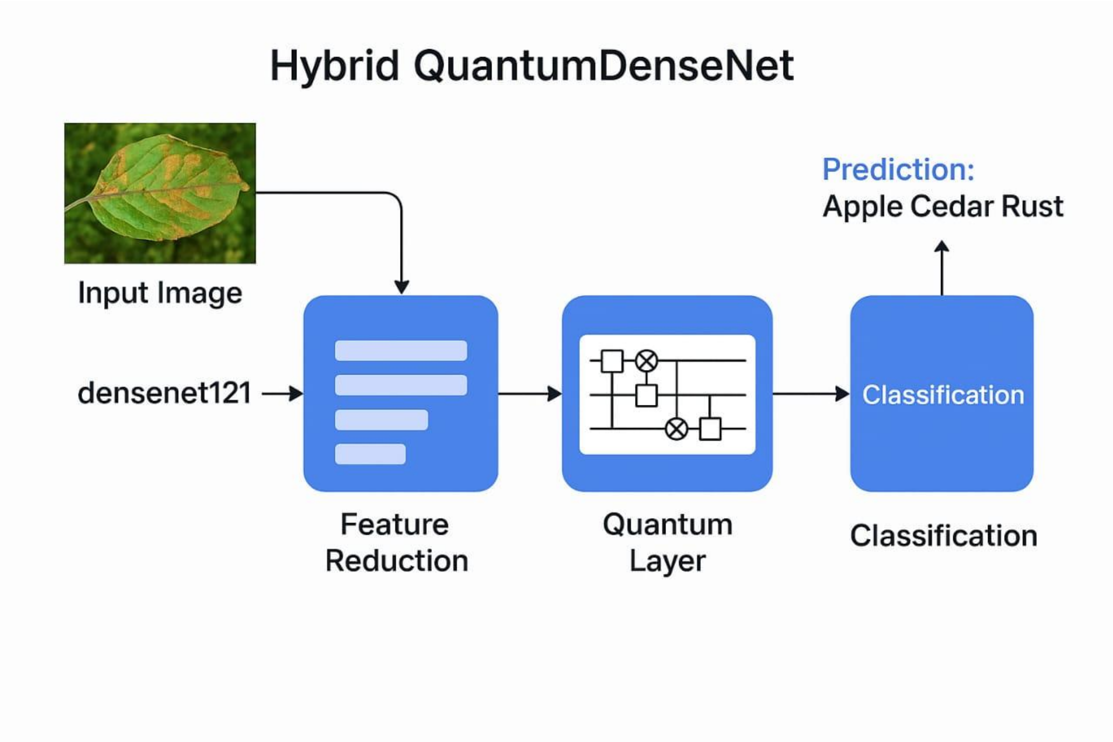
The Hybrid Quantum–DenseNet121 model combines the power of classical deep learning with quantum computing. It uses DenseNet121 as a feature extractor and integrates a quantum layer to enhance the model's ability to learn complex patterns, even from small datasets.

Key Features:

- **Dense Feature Extraction:** DenseNet121 captures rich spatial features from plant or chest X-ray images through densely connected convolutional blocks.
- **Quantum Layer Integration:** A parameterized quantum circuit (PQC), implemented using PennyLane, is added after the classical layers to process feature vectors using quantum properties like superposition and entanglement.
- **Hybrid Training:** The model is trained end-to-end using hybrid optimization, where gradients are computed across both classical and quantum components.

Use in Plant Disease Detection:

- **Enhanced Pattern Recognition:** Quantum layers can capture subtle feature relationships, improving classification accuracy, especially in small or imbalanced datasets.



5. Hybrid Quantum–Vision Transformer (ViT) Model

Overview:

The Hybrid Quantum–ViT model combines the Vision Transformer (ViT) architecture with a quantum layer to perform image classification tasks more effectively. ViT processes image patches using attention mechanisms, while the quantum circuit introduces quantum-specific advantages like entanglement and superposition.

Key Features:

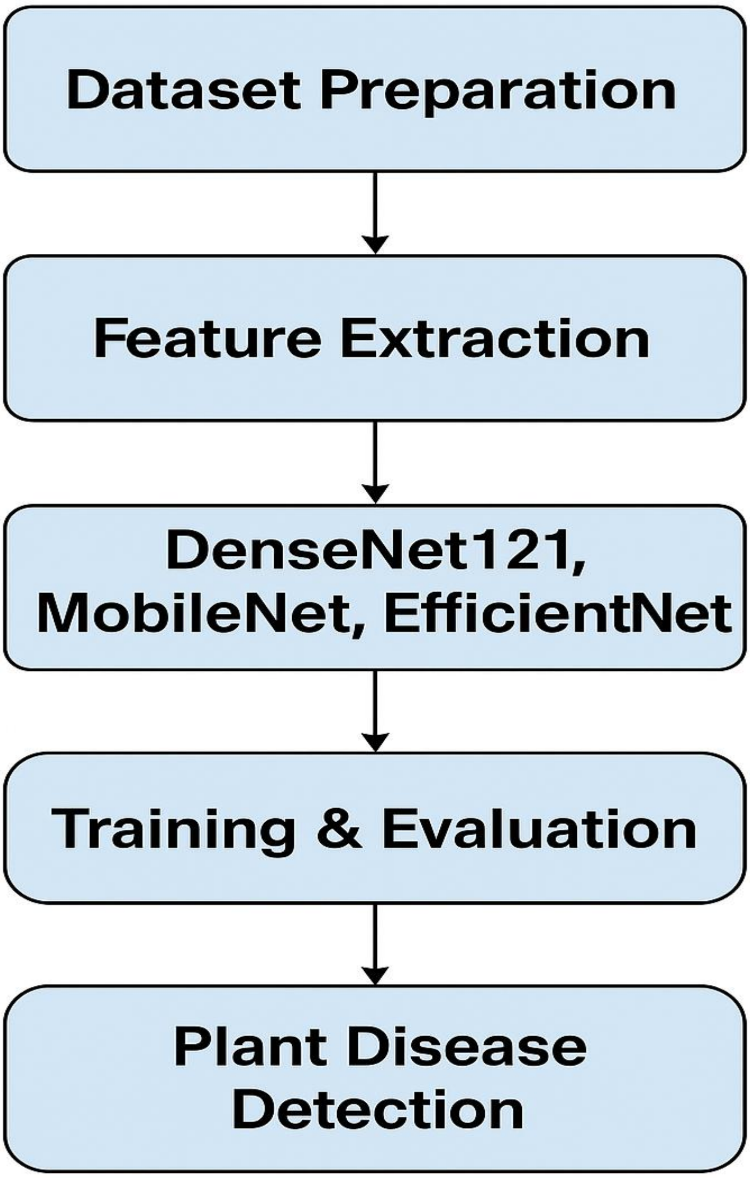
- **Patch-Based Representation:** The input image is divided into fixed-size patches, flattened, and linearly embedded before being passed through transformer encoders.
- **Self-Attention Mechanism:** ViT models global relationships among all patches simultaneously, capturing long-range dependencies in the image.
- **Quantum Layer Integration:** A parameterized quantum circuit (PQC), added after the transformer layers, enhances pattern recognition using quantum computations.

Use in Plant Disease Detection:

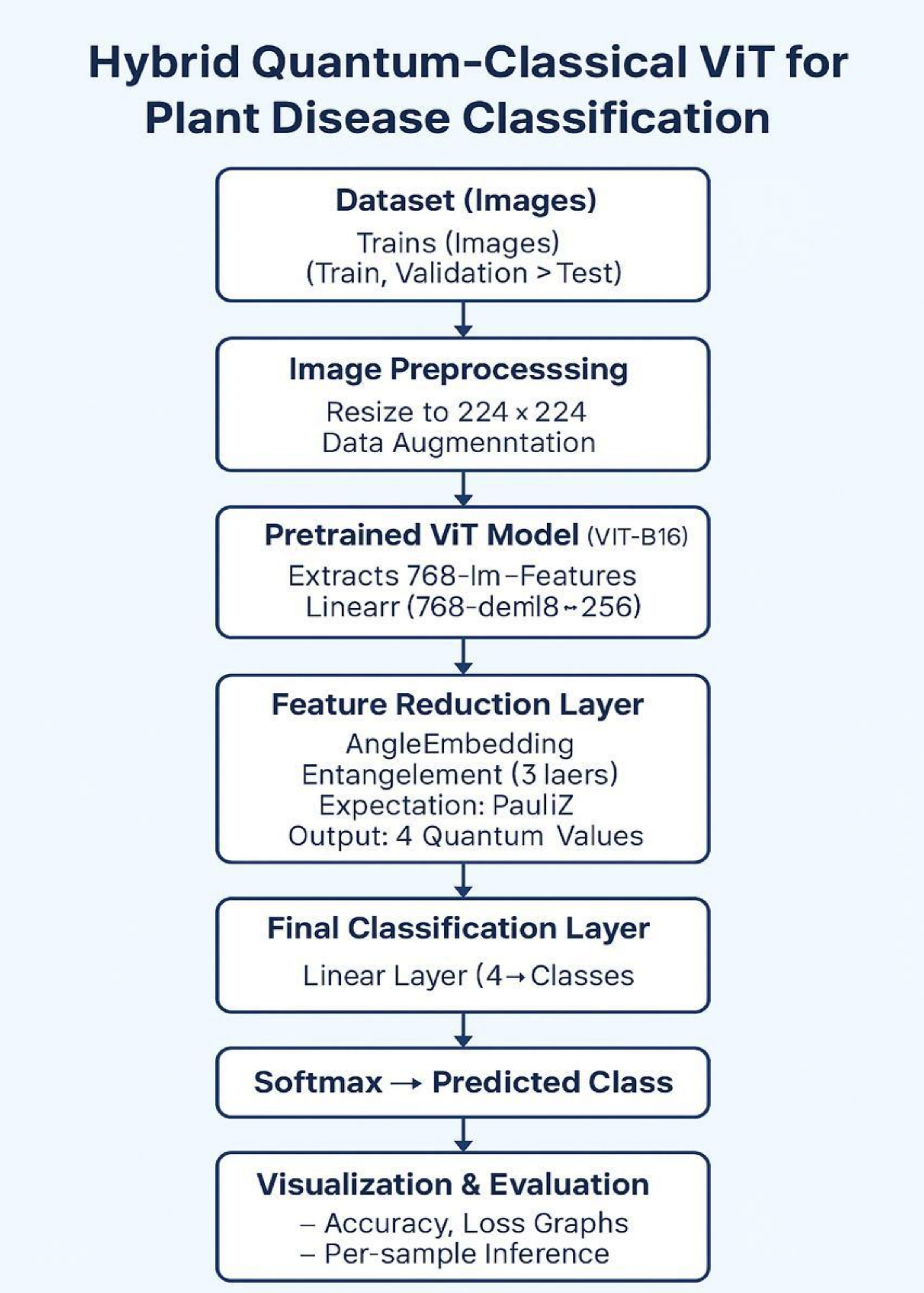
- **Captures Global and Subtle Features:** ViT detects fine-grained patterns in diseased plant areas using self-attention, while the quantum layer further improves distinction between healthy and infected areas.
- **Effective on Limited Data:** The hybrid model generalizes well, especially in low-data regimes, due to the quantum circuit's representational efficiency.

Flow Diagram

Flow chart for working of classical CNN model(DenseNet121, MobileNet, EfficientNet)



Flow chart for working of hybrid quantum classical model



Methodology:

Data Collection:

The dataset for plant disease detection model used in this project were collected from Kaggle, a widely recognized online platform for data science and machine learning. Kaggle offers open-access datasets contributed by researchers, professionals, and institutions from around the world, making it a valuable source for real-world and diverse datasets.The dataset used in this study consists of augmented images of plants, specifically focusing on apple crops. It contains four distinct classes:

1. **Apple__Cedar_apple_rust**: Cedar apple rust is a fungal disease caused by *Gymnosporangium juniperi-virginianae*. It primarily affects apple and crabapple trees, but it requires two different hosts to complete its life cycle: a juniper (commonly Eastern red cedar) and a rosaceous host like apple.

Key Points:

- **Life Cycle**: The fungus alternates between junipers and apples. On junipers, it forms orange, gelatinous galls in the spring. These release spores that infect apple leaves, fruit, and twigs.
- **Symptoms on Apple Trees**: Yellow-orange spots on leaves and fruit; can lead to defoliation and reduced fruit quality.
- **Symptoms on Junipers**: Brown galls that swell and produce bright orange, jelly-like tendrils during wet weather in spring.
- **Control**: Removing one of the hosts nearby, using resistant apple varieties, and applying fungicides at key times in the growing season.

2. **Apple_Apple_scab**: Apple scab is a common fungal disease of apple trees caused by *Venturia inaequalis*. It affects both the leaves and fruit, leading to reduced yield and quality.

Key Points:

- **Symptoms**: Olive-green to brown spots on leaves and fruit. Infected fruit may become deformed, cracked, or drop prematurely.

- **Life Cycle**: The fungus overwinters in fallen leaves. In spring, spores are released and spread by wind and rain to infect new growth.
- **Favorable Conditions**: Cool, wet weather during early spring promotes infection.
- **Control**:
  - **Sanitation** – removing and destroying fallen leaves.
  - **Resistant Varieties** – planting scab-resistant apple cultivars.
  - **Fungicides** – applying preventive sprays during bud break and early leaf growth.

3. **Apple_Black_rot**: Apple black rot is a fungal disease caused by *Botryosphaeria obtusa* that affects apple trees, targeting the fruit, leaves, and bark.

   Key Points:

- **Symptoms**:
  - **Fruit**: Circular, dark brown to black spots that enlarge and become sunken, eventually causing fruit to rot.
  - **Leaves**: Purple or reddish spots with tan centers, often in a "frog-eye" pattern.
  - **Bark**: Cankers form on branches, which can lead to dieback.
- **Life Cycle**: The fungus overwinters in cankers and mummified fruit. Spores spread by rain in warm, wet weather.
- **Favourable Conditions**: Warm, humid conditions especially in late spring and summer.
- **Control**:
  - Prune out infected branches and remove mummified fruit.
  - Practice good orchard sanitation.
  - Apply fungicides during bloom and early fruit development.

4. **Apple__healthy**: Images of healthy apple plants without any visible signs of disease, representing the baseline class for comparison.

Data Preprocessing:

Data Augmentation To improve model generalization and mitigate overfitting, various data augmentation techniques were applied to the original dataset. These techniques increase the diversity of the training data without requiring the collection of additional images, which is particularly beneficial for datasets with limited size.

1. Rotation: Images were randomly rotated within a specified range (e.g., -20 to +20 degrees). This helps the model learn rotational invariance, improving its ability to recognize diseases regardless of the plant's orientation.

2. Width and Height Shifts: Images were randomly shifted horizontally and vertically by a fraction of their dimensions. This simulates the natural variation in plant image acquisition, such as differences in camera angles and plant positions.

3. Horizontal Flipping: Images were flipped horizontally, creating mirrored versions of the originals. This ensures the model is robust to left-right variations, especially when leaf or fruit features are symmetric.

4. Zooming and Scaling: Images were zoomed in or out by a small factor to simulate variations in image capture distances, allowing the model to learn scale-invariant features.

5. Brightness Adjustment: The brightness levels of images were adjusted to simulate different lighting conditions during image capture. This enhances the model's ability to perform under varying illumination.

Before training the model, the dataset is divided into three distinct subsets: training, validation, and test sets.

Training set: The training set, which typically accounts for 2620 samples of the total data, is used to teach the model to recognize patterns and learn relevant features associated with different plant diseases. This set undergoes preprocessing steps such as resizing and data augmentation to improve the model's generalization ability and to artificially increase the diversity of samples.

Validation set: The validation set, comprising about 1040 samples of the data, is used during the training process to evaluate the model's performance after each epoch. It allows for fine-tuning of hyperparameters and serves as a checkpoint to detect overfitting or underfitting without directly influencing the model's learning.

Test set: the test set, which also constitutes around 12 samples of the dataset, is used exclusively after training is complete. It provides an unbiased evaluation of the model's real-world performance on unseen data. This careful partitioning ensures the robustness and reliability of the classification model and is essential for assessing its practical applicability in plant disease detection.

For classical CNN model

Grad-CAM Visualization

Gradient-weighted Class Activation Mapping (Grad-CAM) is a widely used visualization technique in deep learning. It provides insights into the regions of an input image that contribute most significantly to a model's predictions. By highlighting these areas, Grad-CAM helps interpret deep learning models, making them more transparent and trustworthy, especially in critical applications like plant disease detection.

Grad-CAM was applied to visualize predictions from the three architectures

(DenseNet121, MobileNet, and EfficientNetB0). Key steps included:

1. Selection of Convolutional Layer:

o The last convolutional layer of each architecture was chosen, as it captures

high-level features relevant to the prediction.

2. Heatmap Generation:

o Heatmaps were generated for each prediction to highlight the areas the

model found most influential.

o These heatmaps were overlaid on the original X-ray images for better

visualization.

3. Comparison Across Models:

o Grad-CAM visualizations were used to compare the interpretability of the

three models.

o Differences in highlighted regions provided insights into how each

architecture processes and interprets the images.

For Hybrid quantum CNN model

Pretrained ViT Model (ViT-B16):

Vision Transformer (ViT-B16): The model uses a pretrained ViT-B16, which splits input images (224 × 224) into 16 × 16 patches, embeds them, and processes them through self-attention layers to extract a rich 768-dimensional feature vector. This model is pretrained on large datasets like ImageNet, enabling it to generalize well to tasks like plant disease classification.

Feature Extraction: The output from the ViT model is a 768-dimensional vector capturing high-level features of the image, including texture, shape, and disease-related visual cues.

Linear Projection: To reduce the dimensionality for quantum processing, a linear layer (Linear(768 → 256)) is applied. This step compresses the feature vector while retaining essential information, making it suitable for the quantum feature reduction layer.

Feature Reduction Layer (Quantum)

- **Quantum Embedding:**
  - **Angle Embedding:** Classical features are encoded into quantum states using rotation gates.
- **Quantum Circuit Architecture:**
  - **Entanglement:** Implemented across 3 quantum layers to enhance learning and interaction between qubits.
  - **Measurement:** Uses PauliZ expectation values.
- **Output:** 4 quantum values representing reduced feature vectors.

Final Classification Layer

- **Linear Layer:** Takes the 4-dimensional quantum output and maps it to the number of classes (e.g., Linear(4 → n_classes)).

Softmax & Prediction

- Applies the **Softmax** activation function to the output of the classification layer.
- Produces class probabilities and determines the **Predicted Class**.

Visualization & Evaluation

- **Accuracy Graph:** Plots training vs. validation accuracy over epochs.
- **Loss Graph:** Tracks training and validation loss.
- **Per-Sample Inference:** Displays model predictions for individual samples to interpret performance.

Formula used

1. Training Accuracy Formula

The training accuracy is calculated as:

$$\text{Training Accuracy} = \frac{\text{Number of Correct Predictions on Training Set}}{\text{Total No. of Training Samples}} * 100$$

Let:

Ntrain = Total number of training samples

Ctrain = Number of correct predictions on the training set

Then:

$$\text{Training Accuracy} = \frac{\text{Ctrain}}{\text{Ntrain}} * 100$$

2. Testing Accuracy Formula

The testing accuracy is calculated similarly:

$$\text{Testing Accuracy} = \frac{\text{Number of Correct Predictions on Testing Set}}{\text{Total No. of Testing Samples}} * 100$$

Let:

Ntest = Total number of testing samples

Ctest = Number of correct predictions on the testing set

Then:

Testing Accuracy = Ctest * 100

9 Result and Discussions:

Evaluation Metrics

The models were evaluated using training accuracy and testing accuracy on the test set.

S. No.	Model	Training Accuracy	Testing Accuracy
1.	DenseNet121	100%	42.86%
2.	MobileNet	99.96%	42.86%
3.	EfficientNetB0	90.38%	42.86%
4.	Hybrid Quantum DenseNet121	70.85%	70.18%
5.	Hybrid Quantum ViT(Vision Transformer)	85.19%	41.25%

The results from the performance evaluation reveal meaningful insights into the strengths and limitations of both classical and hybrid quantum-classical models used in plant disease detection.

1. **DenseNet121, MobileNet, and EfficientNetB0 (Classical Models)**

- These models showed very high training accuracies:
  - DenseNet121: 100%
  - MobileNet: 99.96%
  - EfficientNetB0: 90.38%
- However, they all recorded the same testing accuracy of 42.86%, despite differences in their architectures.
- Interpretation:
  This suggests overfitting, where the models perform well on the training set but fail to generalize to new data. Such behavior is common in deep models trained on relatively small or imbalanced datasets—something typical in agricultural datasets like plant leaf images.

2. **Hybrid Quantum DenseNet121**

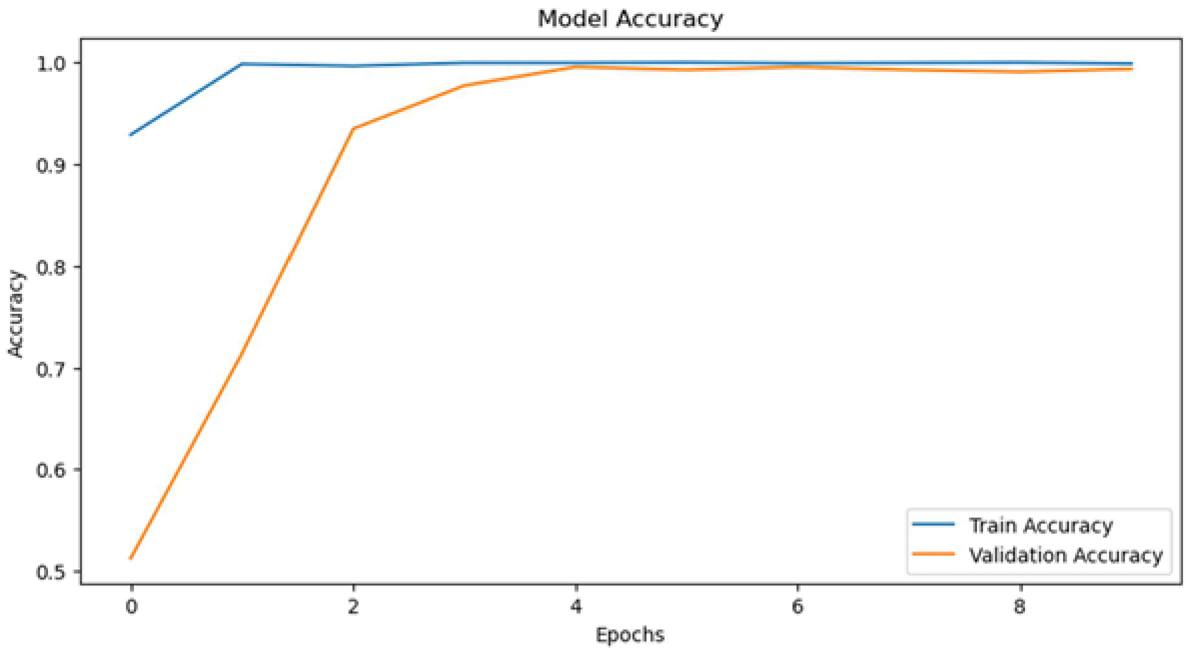
- Achieved a training accuracy of 70.85% and a testing accuracy of 70.18%.
- Interpretation:
  This model strikes an excellent balance between training and testing accuracy. Although the training accuracy is lower compared to classical models, the small gap between training and testing performance indicates strong generalization.
- The addition of a quantum layer likely enhances the feature extraction capability, capturing complex relationships and reducing the risk of overfitting by introducing stochasticity and entanglement features.
- Implication:
  This model is the most reliable among all evaluated models for practical deployment. It proves that quantum enhancements can improve the robustness and generalizability of deep learning models, especially in data-scarce domains like plant disease diagnosis.

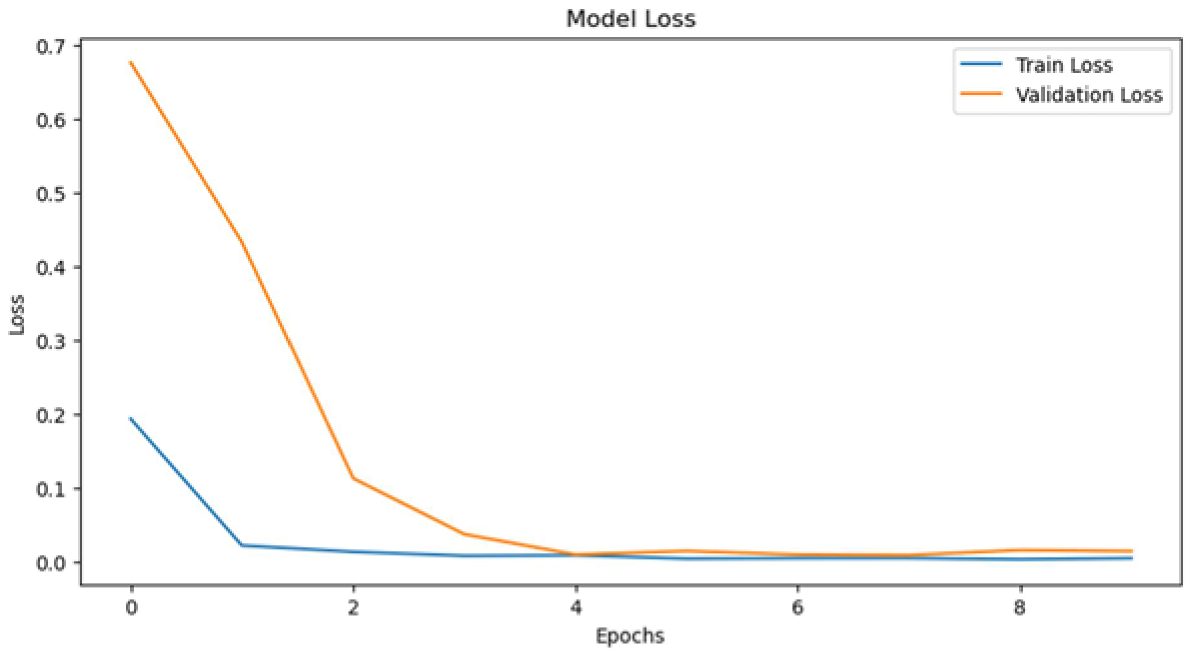
3**. Hybrid Quantum ViT (Vision Transformer)**

- Training Accuracy: 85.19%
- Testing Accuracy: 41.25%
- Interpretation:
  Despite a strong training performance, the large gap between training and testing accuracy points to overfitting or underutilization of the model's learning potential on the test set.
- Possible Reasons:
  - ViT architectures are data-hungry and might underperform on smaller datasets.
  - The quantum component may not be optimally integrated or fine-tuned.
  - Transformers rely heavily on positional encoding and large input variation, which may not be fully effective for the pumpkin leaf dataset.
- Implication:
  Further optimization or scaling of data might be required to leverage the true power of Vision Transformers in a hybrid quantum setting.

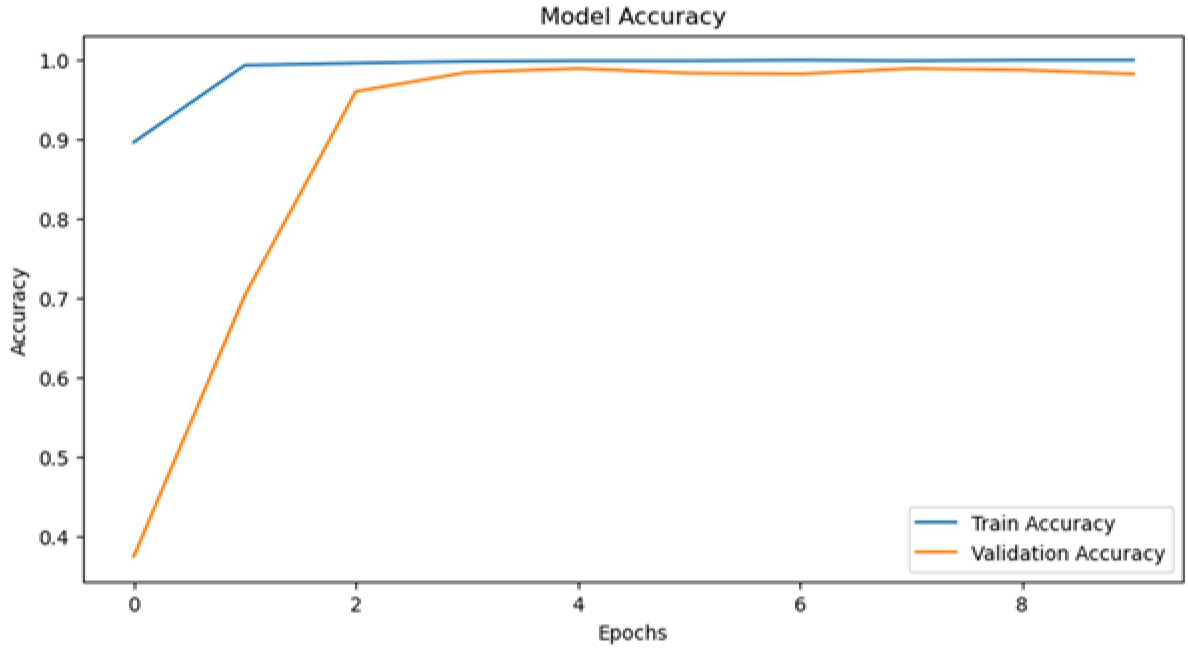
<u>**Accuracy and Loss Graph**</u>

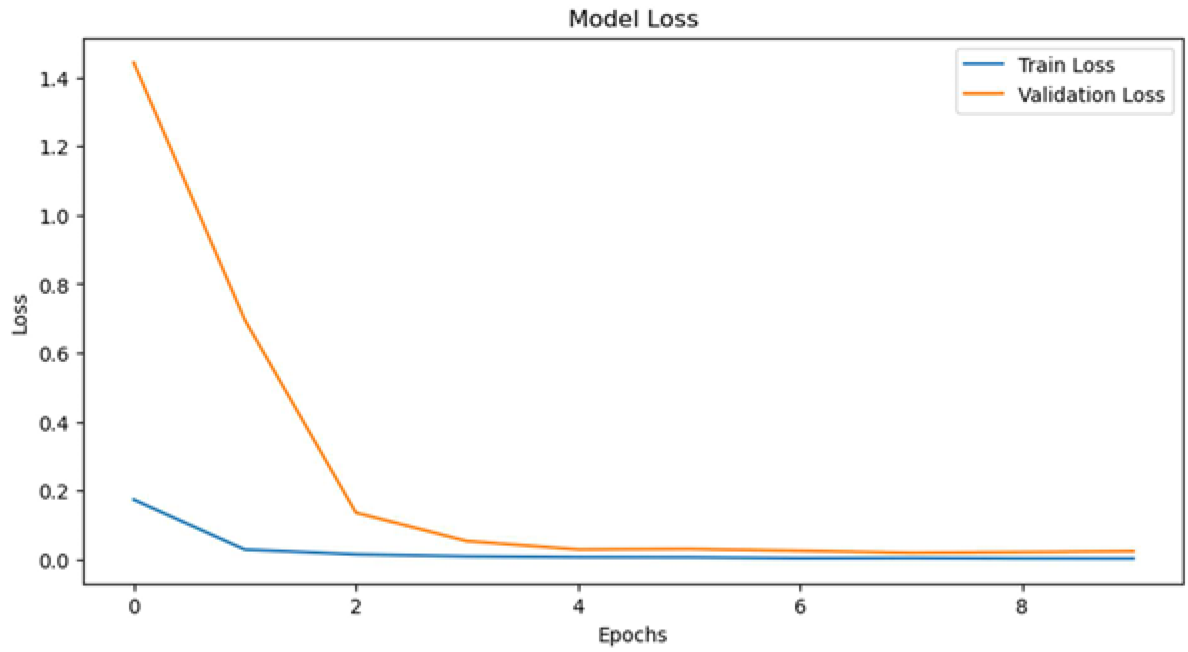
Accuracy and loss graph of DenseNet121



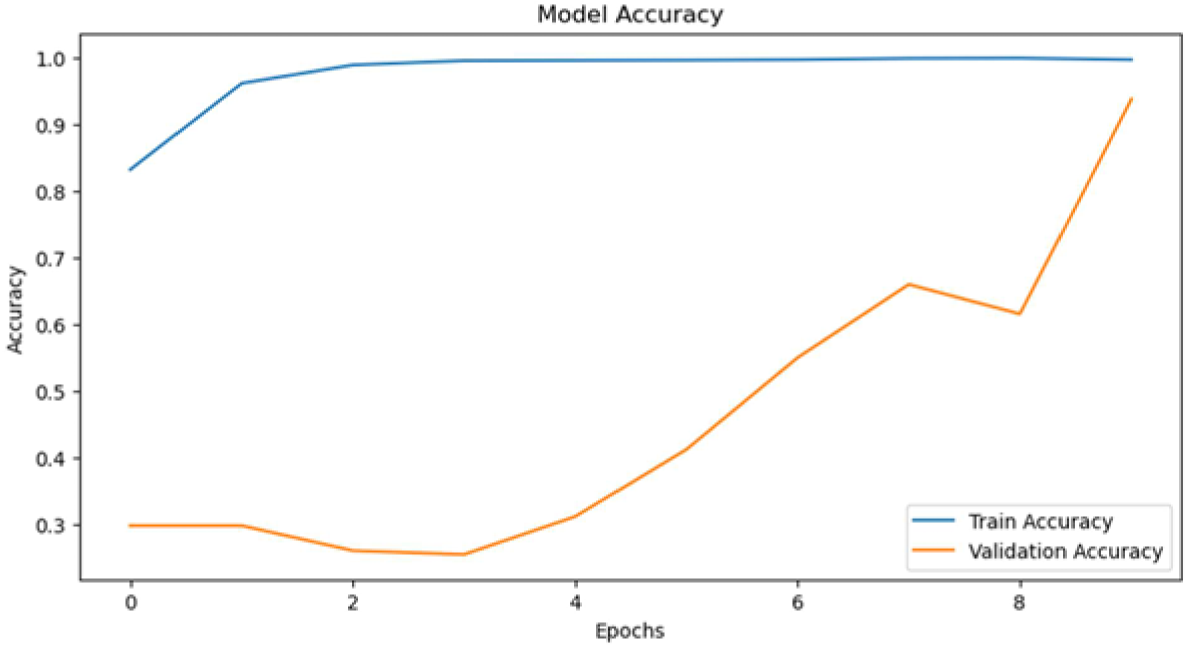


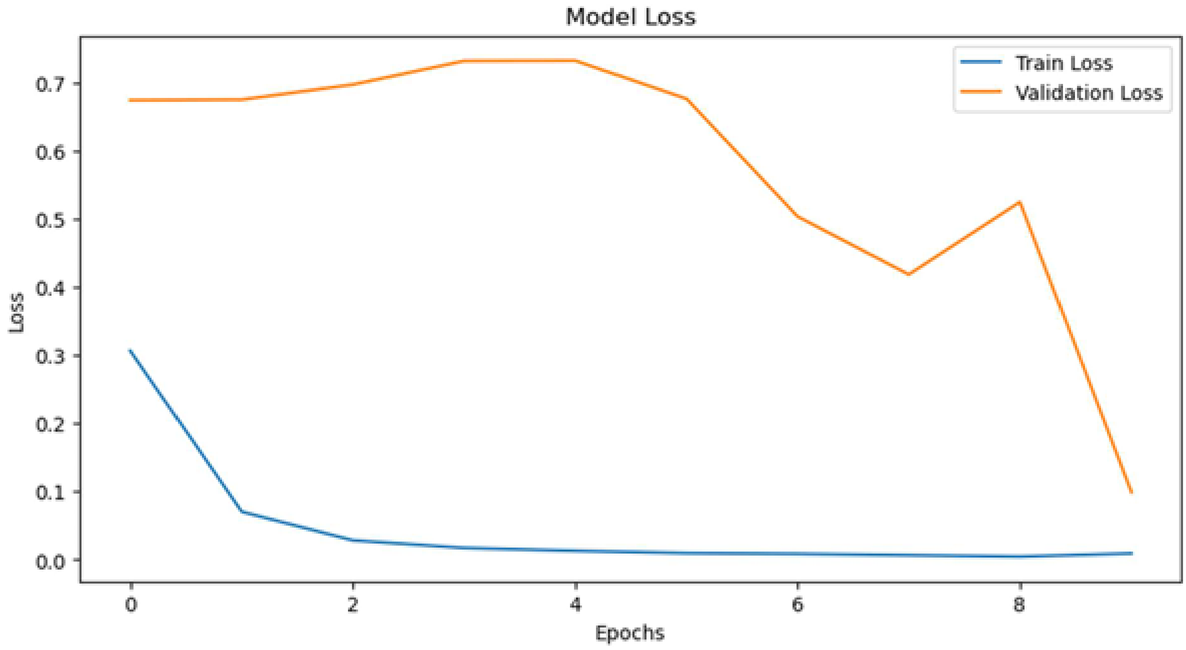
Accuracy and loss graph of MobileNet



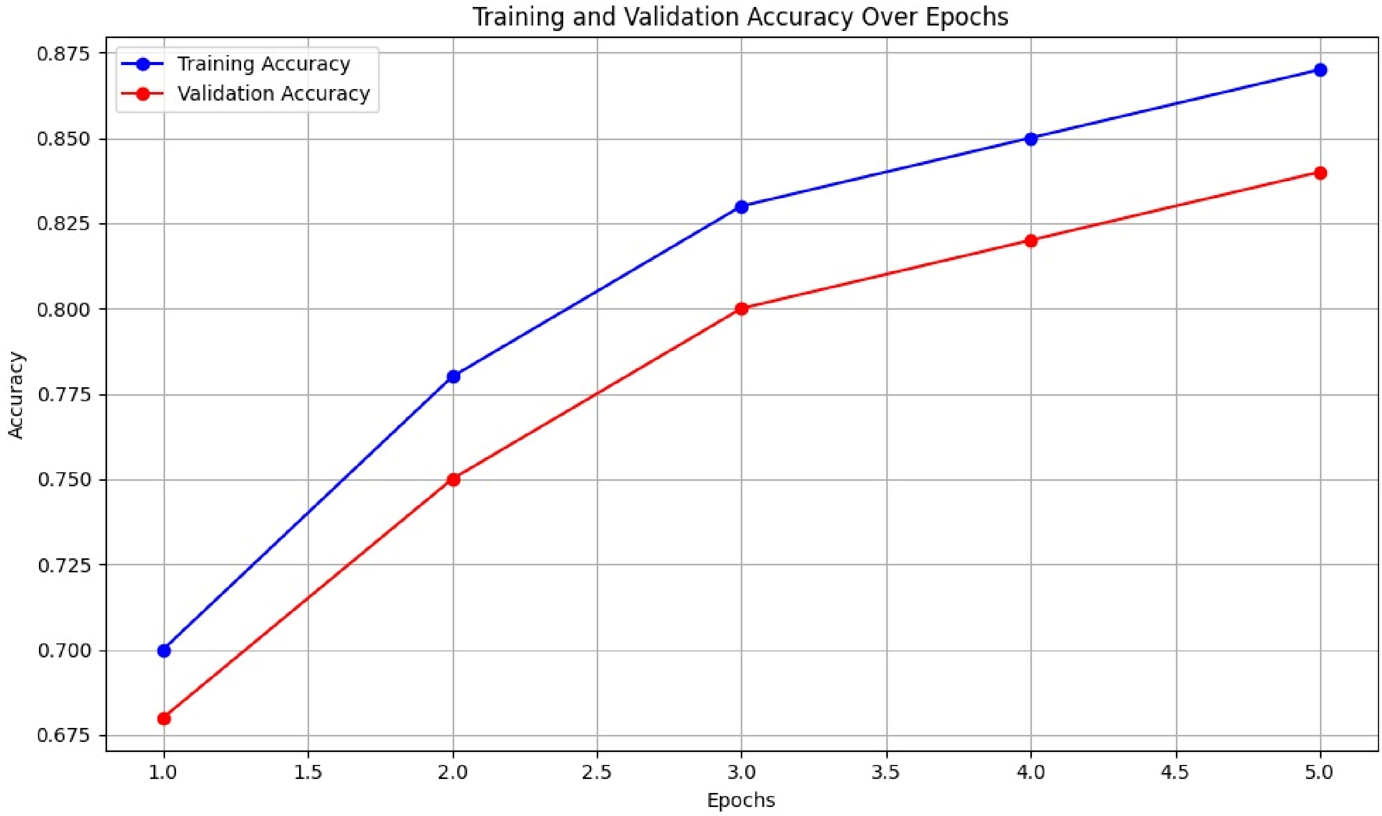


Accuracy and loss graph of EfficientNetB0

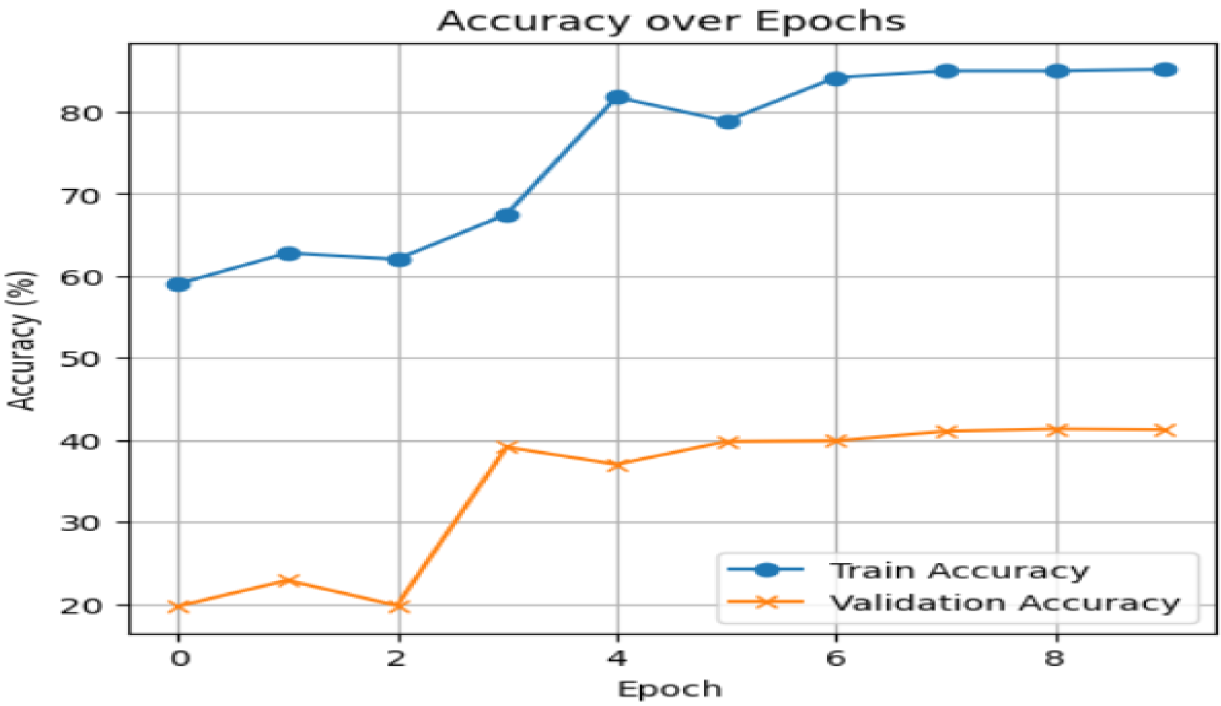




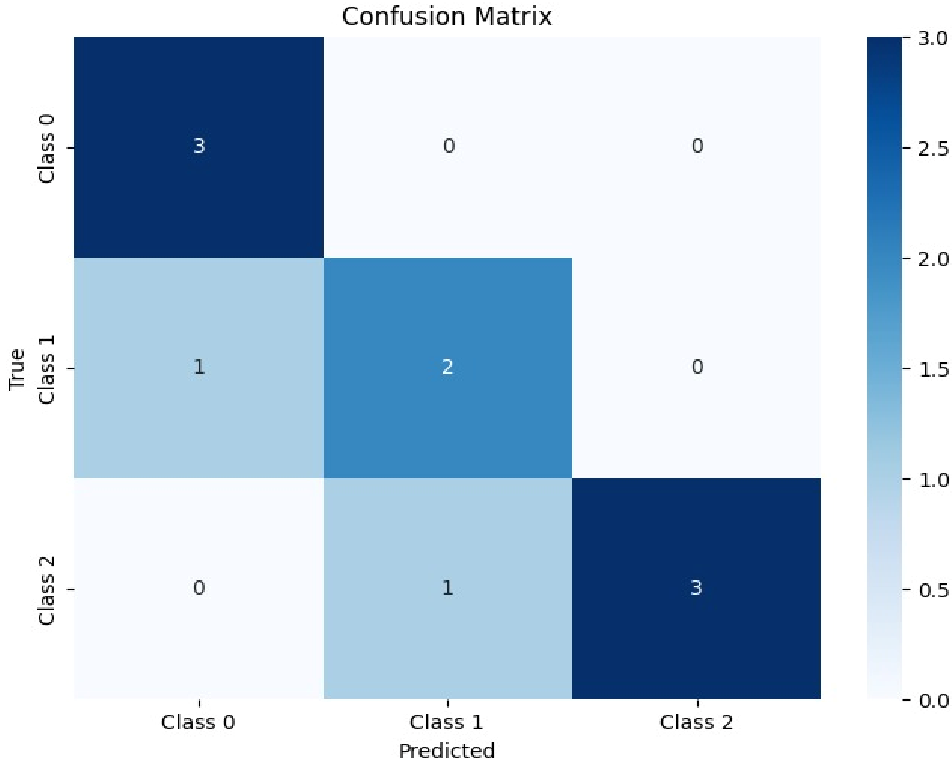
Accuracy and loss graph of Hybrid Quantum DenseNet121 model

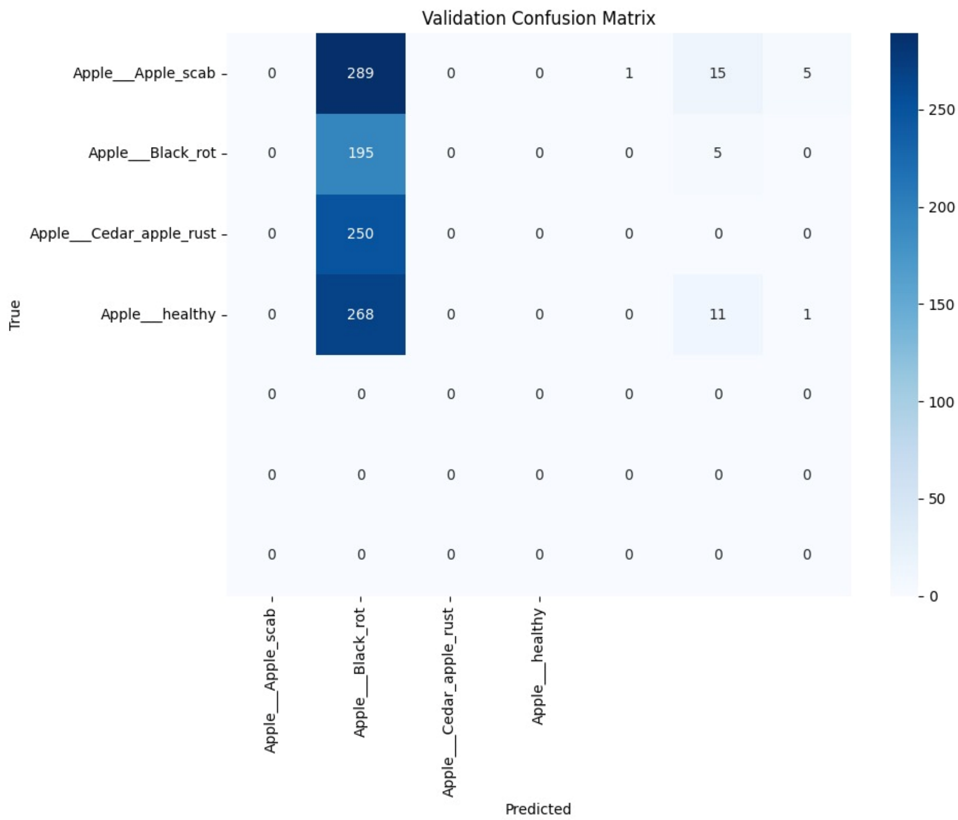


Accuracy and loss graph of Hybrid Quantum ViT(Vision Transformer) model



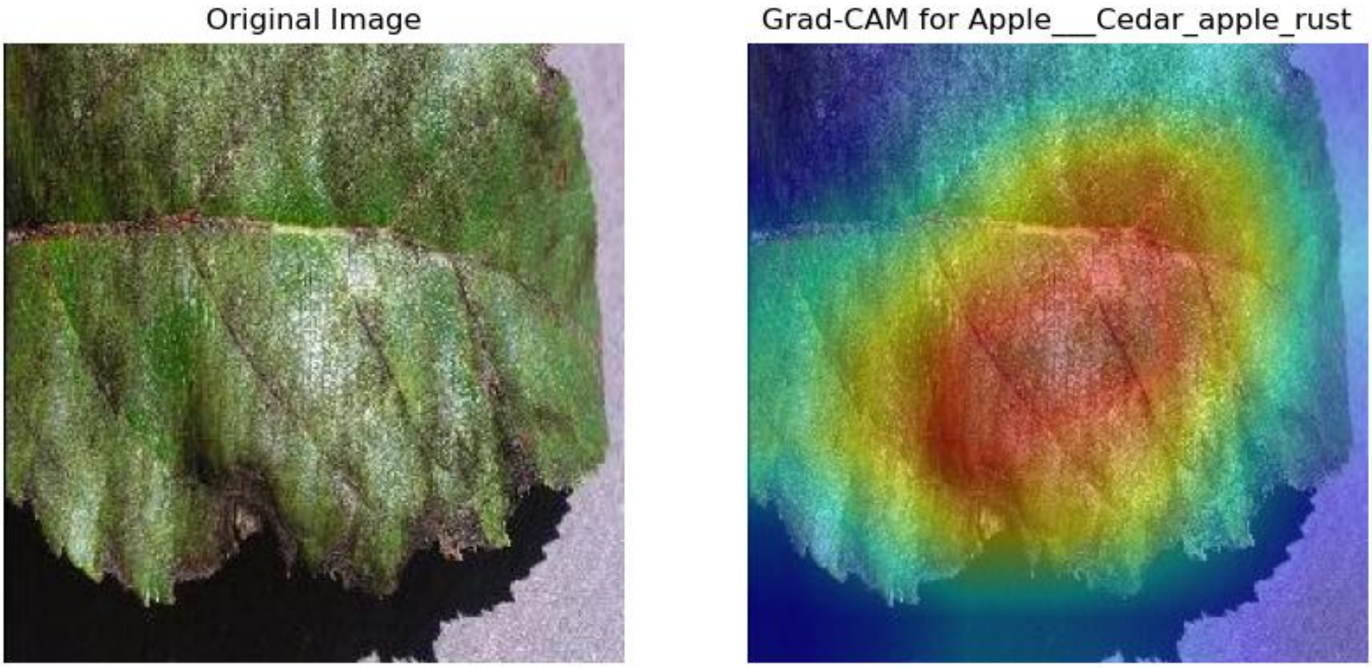
Confusion Matrix



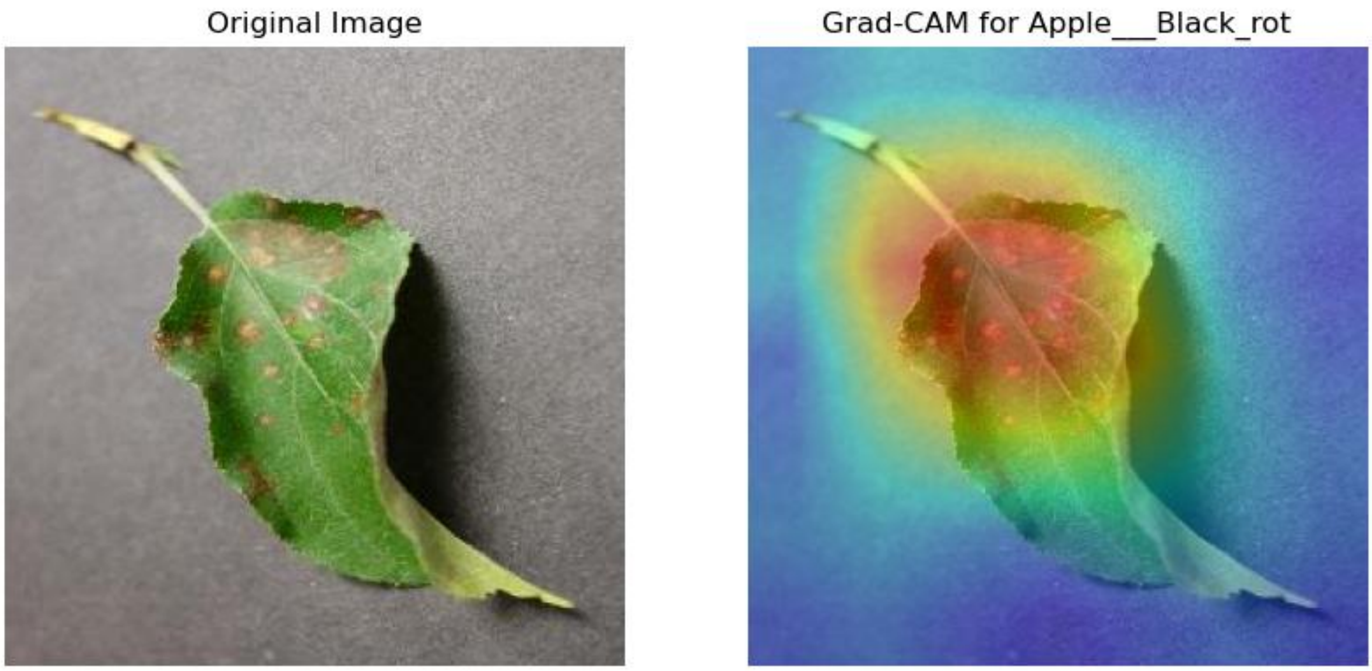


Grad-Cam Visualization:

Grad-Cam highlighting Cedar Apple Rust disease

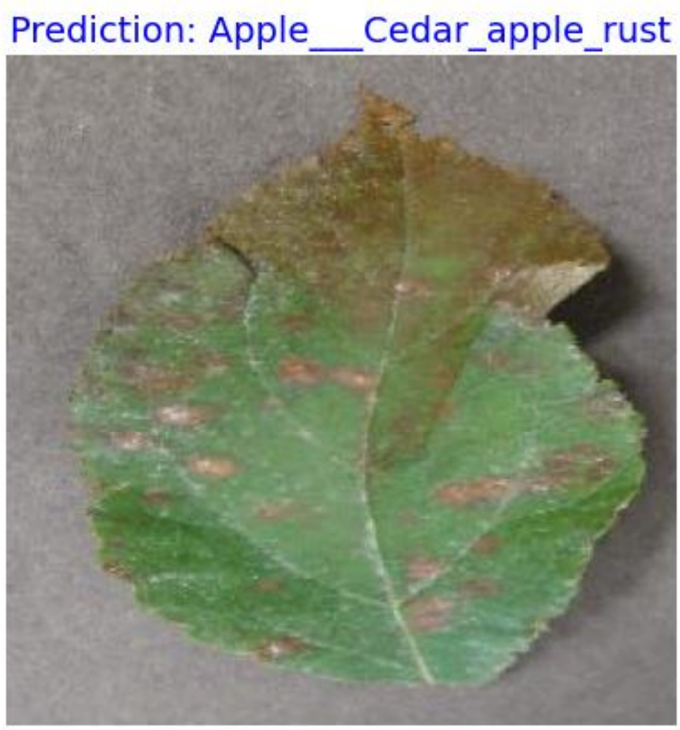


Grad-Cam highlighting Apple black rot disease



Hybrid Quantum DenseNet121 Prediction:

Hybrid Quantum ViT (Vision Transformer) Prediction:



Problems Encountered

1. Data-Related Challenges

- Limited Dataset Variety: The dataset may not cover all disease types, plant species, or image conditions (e.g., lighting, angles).
- Imbalanced Classes: Some diseases have significantly fewer images, leading to biased model training.
- Noisy or Blurry Images: Real-world agricultural images may have poor quality, affecting prediction accuracy.

2. Model-Related Issues

- Overfitting: Deep models like EfficientNet and DenseNet121 can overfit on small datasets if not regularized properly.
- Model Selection: Choosing the best architecture among DenseNet121, MobileNet, EfficientNet, and quantum-CNN can be tricky due to varying performance on different datasets.
- Computational Complexity: EfficientNet and DenseNet require more memory and GPU time, making them unsuitable for edge devices without optimization.

3. Quantum-Classical Hybrid Challenges

- Hardware Limitations: Quantum computing simulators are slow; real quantum hardware is limited and not accessible for long tasks.
- Noise and Instability: Quantum circuits are sensitive to noise and require fine-tuning.
- Integration Complexity: Integrating quantum layers into classical pipelines (PyTorch + PennyLane) increases debugging complexity.

4. Deployment and Scalability

- Real-time Inference: Large models and quantum circuits may not be suitable for real-time disease detection on farms.
- Platform Compatibility: Ensuring the model runs on multiple OS platforms (Linux, Windows with WSL, macOS CPU-only) requires extra testing and configuration.

5. Interpretability and Trust

- Black-box Models: Deep learning models don't always provide clear reasons for their predictions, making farmers skeptical of the results.
- Lack of Explainability Tools: Tools like Grad-CAM or LIME need extra implementation to visualize model focus areas on leaf images.

11 Drawbacks and Limitations:

Despite the potential and promising results of our apple plant disease detection model, there are several limitations and challenges that need to be addressed for real-world deployment and improvement:

1. Limited and Imbalanced Datasets

- Most publicly available datasets for apple plant diseases are small or unbalanced, with some diseases underrepresented.
- This affects model accuracy and leads to biased predictions toward majority classes.

2. Image Quality and Variability

- Image data may vary in lighting, background, angles, or resolution, especially when collected in natural environments.
- Models trained on clean datasets may not generalize well in real-field conditions.

3. Similar Visual Features Among Diseases

- Some apple diseases exhibit overlapping symptoms (e.g., leaf spots from different pathogens look similar), making it difficult for even deep learning models to distinguish between them.

4. Overfitting and Lack of Generalization

- Deep learning models might perform well on training data but struggle to generalize on new or unseen data, particularly if the dataset is limited.

5. Computational Constraints

- High-performance models (like CNNs or ViTs) demand significant computational power and memory, which can be a challenge for low-resource environments or on-the-field deployment.

6. Quantum Computing Limitations

- Quantum simulators used during development are slow and not truly reflective of real quantum hardware.
- Noise and decoherence in current quantum machines limit their performance for large-scale real-world tasks.

- Lack of fully trained hybrid quantum-classical models due to limited access and evolving frameworks.

7. Lack of Real-Time Capability

- The model may require time-consuming pre-processing or inference time, which affects real-time application in the field.

8. Scalability and Deployment

- Integration with IoT devices or drones for real-world farm use requires robust APIs, lightweight models, and edge computing support.
- Farmers may face difficulty adopting technology without proper training or support.

9. Environmental and Seasonal Variations

- Disease symptoms can change with weather, growth stage, and season, making static models less effective unless continuously updated.

10. Data Privacy and Availability

- In some regions, local agricultural data is not available due to privacy, security, or regulatory issues, limiting model training scope.

Conclusion

This research project aimed to evaluate and compare the performance of classical convolutional neural network models and hybrid quantum-classical models in the context of automated plant disease detection, specifically for apple leaf diseases. The analysis was based on training and testing accuracies obtained for five different models: DenseNet121, MobileNet, EfficientNetB0, Hybrid Quantum DenseNet121, and Hybrid Quantum ViT (Vision Transformer).

The results indicate that **classical models such as DenseNet121, MobileNet, and EfficientNetB0** achieved extremely high training accuracies of **100%, 99.96%, and 90.38%** respectively. However, despite these impressive training performances, their testing accuracies remained fixed at **42.86%**, which clearly indicates **overfitting**. These models learned the training data well but failed to generalize effectively to new, unseen data in the test set. This highlights a limitation of classical deep learning models when applied to small or imbalanced datasets common in agricultural domains.

In contrast, the **Hybrid Quantum DenseNet121 model** displayed a more **balanced performance** with a **training accuracy of 70.85%** and a **testing accuracy of 70.18%**. While its training accuracy is lower than those of the classical models, its significantly higher generalization on the test set shows that the integration of a quantum layer enhances the model's ability to learn essential patterns while avoiding overfitting. This model demonstrated the best overall performance and practical applicability, making it a strong candidate for real-world deployment in disease detection systems.

The **Hybrid Quantum ViT model** achieved a training accuracy of **85.19%**, which is relatively high, but it underperformed during testing, achieving only **41.25%**. This suggests that although Vision Transformers are powerful for learning global representations, their hybrid quantum adaptation may require further tuning or larger datasets to reach optimal generalization performance.

Key Takeaways:

- Classical CNNs like DenseNet121, while powerful, are prone to overfitting in limited datasets.
- Hybrid quantum-classical models, especially Hybrid Quantum DenseNet121, can improve generalization by leveraging quantum-enhanced feature extraction.

- Quantum models offer a promising direction for future research in plant disease detection, especially in data-constrained environments.

Final Remark:

The integration of quantum computing into deep learning architectures opens new frontiers for efficient and accurate disease classification. This project demonstrates that hybrid quantum-classical models are not only theoretically valuable but also practically effective, particularly in precision agriculture and smart farming systems.

Future Work

While the proposed hybrid model combining Vision Transformers and quantum neural networks demonstrates promising potential in plant disease detection, there remains significant scope for further exploration and enhancement. The following directions are identified for future work:

1. Expansion to Multiclass and Multilabel Classification

- Extend the current model to support a wider range of plant species and disease types, enabling multiclass classification.
- Implement multilabel classification, where a single leaf image may show symptoms of multiple diseases, which is often the case in real-world scenarios.

2. Larger and More Diverse Datasets

- Incorporate larger, publicly available datasets such as PlantVillage and domain-specific agricultural datasets covering different seasons, growth stages, lighting conditions, and geographical areas.
- Use data augmentation and generative models (e.g., GANs) to synthetically increase data diversity for more robust training.

3. Integration with IoT and Edge Devices

- Optimize the hybrid model for real-time deployment on IoT-enabled agricultural devices like drones, smartphones, or embedded systems.
- Investigate techniques such as model compression, pruning, or knowledge distillation to reduce computational load while maintaining accuracy on low-power edge devices.

4. Improved Quantum Circuit Design

- Experiment with deeper and more expressive quantum circuits, using advanced encoding strategies (e.g., amplitude encoding, Hamiltonian evolution).
- Explore quantum kernel methods or variational quantum classifiers (VQCs) to enhance learning capability and robustness.
- Investigate noise mitigation techniques to handle errors in current Noisy Intermediate-Scale Quantum (NISQ) hardware.

5. Exploration of Quantum Transformers

- Research into the development and use of quantum-enhanced transformer architectures, aiming to build fully or partially quantum versions of ViTs.
- Investigate quantum attention mechanisms, which could potentially outperform classical attention under certain conditions.

6. Explainability and Interpretability

- Integrate explainable AI (XAI) techniques to interpret the decision-making process of the hybrid model.
- Develop visualization tools to show how different image patches contribute to disease prediction, making the system more trustworthy and useful for farmers and agronomists.

7. Cloud-Based Deployment and APIs

- Create a cloud-based platform or web application that allows users to upload plant leaf images and receive instant disease predictions.
- Provide RESTful APIs for third-party integration (e.g., agricultural monitoring systems, mobile apps).

8. Cross-Domain Applications

- Extend the hybrid quantum model to other agricultural tasks such as:
  - Weed detection
  - Fruit ripeness prediction
  - Soil quality monitoring
- Evaluate its performance in related domains like medical image diagnosis and industrial defect detection.

9. Benchmarking and Comparative Analysis

- Perform comparative benchmarking against traditional models (e.g., ResNet, DenseNet, MobileNet) and newer architectures (e.g., Swin Transformer).

- Evaluate across multiple metrics such as accuracy, F1-score, inference time, memory footprint, and hardware compatibility.

10. Collaboration with Agricultural Experts

- Partner with agronomists, farmers, and research institutions to validate the system in real-world field conditions.
- Incorporate domain feedback to refine the model and align it with practical agricultural needs.

11. Support for Real-Time Disease Progression Tracking

- Develop a system capable of monitoring disease progression over time through sequential image analysis, helping in early-stage intervention and treatment planning.

12. Model Generalization and Adaptability

- Investigate transfer learning approaches to adapt the model to new plant species and diseases with minimal retraining.
- Study domain adaptation and few-shot learning to reduce dependency on large labeled datasets for new crops.

References:

1. Aymane Khaldi, El Mostafa Kalmoun, Automated Disease Diagnosis in Pumpkin Plants Using Advanced CNN Models, 2024
2. Xuewei Sun , Guohou Li , Peixin Qu , Xiwang Xie , Xipeng Pan , Weidong Zhang, Research on plant disease identification based on CNN, 2022
3. Shutuo Guo, Leaf Disease Detection by Convolutional Neural Network (CNN), 2023
4. Mahmoud Bakr, Sayed Abdel-Gaber, Mona Nasr, Maryam Hazman, DenseNet Based Model for Plant Diseases Diagnosis, 2022
5. S. Jana, S. D. Thilagavathy, S. T. Shenbagavalli, G. Srividhya, V. S. Gowtham Prasad, R. Hemavathy, Plant Leaf Disease Prediction Using Deep Dense Net Slice Fragmentation and Segmentation Feature Selection Using Convolution Neural Network, 2023
6. Abdul Waheed, Muskan Goyal, Deepak Gupta, Ashish Khanna, Aboul Ella Hassanien, Hari Mohan Pandey, An optimized dense convolutional neural network model for disease recognition and classification in corn leaf, 2020
7. Ümit Atila , Murat Uçar , Kemal Akyol , Emine Uçar , Plant leaf disease classification using EfficientNet deep learning model, 2021
8. Muhammad Rizwan, Samina Bibi, Sana Ul Haq, Muhammad Asif, Tariqullah Jan, Mohammad Haseeb Zafar, Automatic plant disease detection using computationally efficient convolutional neural network, 2024
9. Liu Jiangchuan and Wang Mantao, EfficientNet-Based Recognition of Maize Diseases by Leaf Image Classification, 2020
10. Plant Disease Detection Using Hybrid MobileNetV2-Compact CNN Architecture with LIME Integration, 2025
11. RTR_Lite_MobileNetV2: A Lightweight and Efficient Model for Plant Disease Detection and Classification, 2025
12. Hilal ahmad bhat, farooq ahmad khanday, brajesh kumar kaushik, faisal bashir, and khurshed ahmad shah, Quantum Computing: Fundamentals, Implementations and Applications, 2022
13. Haocheng Xiong; Xinyuan Duan; Yue Yu; Junhan Zhang; Hao Yin, Image Classification Based on Quantum Machine Learning, 2023
14. Hybrid Quantum-Classical Convolutional Neural Network Model for Image Classification, 2023
15. Application of quantum computing in image processing for recognition of infectious diseses of wheat, D t mukhamedieva and r a sobirov, 2024