Hybrid Deep Learning Approach for Sweet Orange Leaf Disease Detection Using CNN and Vision Transformers

By
Nahidul Islam
201-15-3153

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by
Mr. Mahimul Islam Nadim
Lecturer
Department of Computer Science and Engineering, Daffodil International University

Co-Supervised by
Ms. Aliza Ahmed Khan
Lecturer (Senior Scale)
Department of Computer Science and Engineering, Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

November 12, 2024

APPROVAL

This Project titled **"Hybrid Deep Learning Approach for Sweet Orange Leaf Disease Detection Using CNN and Vision Transformers"**, submitted by Nahidul Islam, ID No: **201-15-3153** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 13 January, 2025.

BOARD OF EXAMINERS

Dr. Sheak Rashed Haider Noori — **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Most. Hasna Hena — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Md. Ferdouse Ahmed Foysal — **Internal Examiner**
Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Md. Arshad Ali — **External Examiner**
Professor
Department of Computer Science and Engineering
Hajee Mohammad Danesh Science and Technology University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mr. Mahimul Islam Nadim, Lecturer**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this projectnor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

11/01/25

Mr. Mahimul Islam Nadim
Lecturer
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Ms. Aliza Ahmed Khan
Lecturer (Senior Scale)
Department of Computer Science and Engineering
Daffodil International University

Submitted by:

Nahidul

Nahidul Islam
Student ID: 201-15-3153
Department of Computer Science and Engineering
Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the Final Year Design Project (FYDP) successfully.

We are grateful and wish our profound indebtedness to Mr. Mahimul Islam Nadim, Lecturer, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of Deep Learning carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

Sweet orange leaf diseases significantly threaten agriculture, necessitating accurate and timely detection for sustainable farming. This study presents a hybrid deep learning approach combining Vision Transformers (ViT) and Convolutional Neural Networks (CNN) for classifying sweet orange leaf diseases. The methodology includes data preprocessing, such as resizing, normalization, and augmentation, to enhance dataset quality and prepare it for deep learning models. Three models—ViT, ResNet50v2, and the hybrid ViT-CNN—were implemented and evaluated. The hybrid ViT-CNN model achieved the highest test accuracy of 98%, surpassing the individual performances of ViT (90%) and ResNet50v2 (97%), with consistent training and validation accuracies of 97%. The hybrid model integrates the localized feature extraction of CNNs with the global contextual capabilities of ViTs, enabling superior disease classification. This research highlights the scalability and robustness of the hybrid approach, addressing dataset scarcity and computational efficiency challenges. Implemented on Google Colaboratory, the system is optimized for deployment in resource-constrained environments, ensuring accessibility for small-scale farmers. The findings contribute to precision agriculture by reducing crop losses, minimizing pesticide use, and promoting sustainable practices. This study establishes a reliable framework for agricultural disease detection, paving the way for advancements in AI-driven solutions for broader crop management applications.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides an overview of the importance of sweet orange disease detection's importance and the challenges traditional methods face. It highlights the advancements in deep learning, including CNNs, Vision Transformers (ViTs), and hybrid models, and their potential for addressing these challenges.

1.1 Introduction

Plant diseases continue to pose a substantial threat to global agriculture, significantly affecting crop yields, food security, and economic stability. Among these, sweet orange, a vital citrus crop, is highly vulnerable to various diseases, which, if left unchecked, can result in severe economic losses. The accurate and timely detection of these diseases is crucial to enable proactive interventions and safeguard crop health. However, traditional manual methods for disease detection are labor-intensive, subjective, and often error-prone, particularly in large-scale agricultural operations [1]. These limitations underscore the need for automated, scalable, and accurate solutions for disease identification.

Deep learning models, particularly Convolutional Neural Networks (CNNs), have emerged as powerful tools for image-based disease detection, owing to their ability to extract and classify complex features. For example, Khattak et al. (2021) reported a test accuracy of 94.55% using a CNN model for classifying healthy and diseased citrus fruits and leaves. Similarly, Lanjewar and Parab [2] achieved 98% accuracy by leveraging transfer learning with CNNs for citrus leaf disease classification. While these findings highlight the utility of CNNs in agricultural applications, challenges such as generalization across diverse datasets and scalability persist.

The advent of Vision Transformers (ViTs) has revolutionized plant disease detection by enabling superior feature representation and capturing global contextual information. Recent studies demonstrate the effectiveness of ViTs in achieving high accuracy in various agricultural tasks. Boukabouya [3] achieved 99.7% accuracy in

detecting tomato leaf diseases, while Dümen [4] reported 99.84% accuracy in lemon quality classification. These results underscore the potential of ViTs for agricultural disease classification. However, the high computational requirements of ViTs can limit their adoption in resource-constrained environments.

Hybrid models, combining CNNs and ViTs, represent a promising avenue for addressing these limitations. By leveraging the local feature extraction capability of CNNs and the global contextual understanding of ViTs, hybrid models can achieve improved accuracy and scalability. For instance, Thakur [5] proposed the PlantViT model, which achieved 98.61% accuracy on the PlantVillage dataset, while Yong [6] highlighted the practicality of hybrid architectures for plantation use. Despite these advancements, challenges such as dataset scarcity, real-world validation, and explainability remain, hindering the broader deployment of these models [8], [11].

In this context, our study focuses on leveraging CNNs, ViTs, and their hybrid architectures to develop a robust model for sweet orange leaf disease detection. By systematically comparing the performance of these models, we aim to contribute to the development of scalable and interpretable solutions for sustainable agriculture.

1.2 Motivation

The agricultural sector continues to face significant challenges from plant diseases, including those affecting sweet orange crops. These diseases not only threaten crop yields but also contribute to economic strain and exacerbate food insecurity. Traditional methods of disease identification, which often rely on human expertise, are inherently limited by subjectivity, labor intensity, and scalability issues [1]. As a result, there is a pressing need for automated solutions that enhance the efficiency and accuracy of disease detection.

Deep learning has proven to be a transformative technology in addressing these challenges. While CNNs have demonstrated their efficacy in extracting and classifying image features, their performance is often hindered by the need for large, high-quality datasets and challenges in generalization across diverse disease types [2], [9]. ViTs, on the other hand, have shown remarkable potential in overcoming these limitations by offering enhanced feature representation and global contextual understanding.

Studies by Boukabouya [3] and Dümen[4] illustrate the superior performance of ViTs in plant disease detection. However, their computational demands make them less accessible for small-scale farms and resource-constrained settings.

Hybrid models, such as ViT-CNN architectures, offer a balanced solution by combining the strengths of CNNs and ViTs. These models not only achieve high accuracy but also provide scalability and efficiency for real-world applications. Thakur [5] and Yong [6] have highlighted the potential of these architectures in agricultural contexts. Nevertheless, the scarcity of comprehensive datasets for sweet orange leaf diseases and the lack of interpretability features in many models remain critical obstacles [6], [10]. Additionally, most studies rely on controlled datasets, failing to address the variability and unpredictability of field conditions [11].

Motivated by these gaps, our research aims to develop a hybrid ViT-CNN model tailored for sweet orange leaf disease detection. By addressing the challenges of dataset quality, real-world adaptability, and model interpretability, this study seeks to contribute to the growing body of knowledge in agricultural disease detection and support the adoption of AI-driven solutions for sustainable farming.

1.3 Objectives

The primary aim of this thesis is to develop and evaluate deep learning models for the detection and classification of sweet orange leaf diseases. The study focuses on implementing three advanced architectures: Vision Transformers (ViT), Convolutional Neural Networks (CNN), and a hybrid CNN-ViT model. Each model is tested and compared in terms of accuracy and performance to identify the most effective approach for disease detection. The hybrid CNN-ViT model demonstrated superior results, underscoring its potential as a robust solution. Preprocessing techniques, including augmentation and contrast enhancement, were also applied to improve dataset quality and optimize model performance. Additionally, the research includes a comprehensive analysis of the results to assess the strengths and limitations of each model and validate their applicability in real-world scenarios.

- To implement Vision Transformers (ViT), Convolutional Neural Networks (CNN), and a hybrid CNN-ViT model for detecting sweet orange leaf diseases.

- To compare the accuracy and performance of these models, emphasizing the superior results of the hybrid CNN-ViT model.

- To preprocess the dataset with techniques such as augmentation and contrast enhancement to improve the training and testing process.

- To evaluate and analyze the results of each model to identify their strengths and limitations.

- To provide insights into the practical applicability of the hybrid CNN-ViT model for real-world sweet orange leaf disease detection.

1.4 Methodology

The methodology of this research involves the systematic development, implementation, and evaluation of three deep learning architectures: Vision Transformers (ViT), Convolutional Neural Networks (CNN), and a hybrid CNN-ViT model for the detection of sweet orange leaf diseases.

The process begins with data preparation, where the dataset is preprocessed to enhance its quality. Preprocessing techniques include data augmentation (e.g., rotation, flipping, and zooming) to increase dataset variability and improve model generalization, along with contrast enhancement to ensure clear feature representation. These steps help address dataset limitations and improve the training process.

Each model is developed and trained on the preprocessed dataset. The ViT model is used to leverage its ability to capture global contextual information, while the CNN model is applied to explore its capability for extracting localized features. The hybrid CNN-ViT model combines these strengths, using CNN layers to extract low-level features and ViT layers to process and classify the global features.

The training and testing processes are performed using consistent datasets, with metrics such as accuracy serving as the primary evaluation criterion.

Hyperparameters for all models are fine-tuned to achieve the best performance. The results of ViT and CNN models are analyzed and compared before implementing the hybrid model. The hybrid model is designed to build on the findings from the standalone models, aiming to achieve higher accuracy and better overall performance.

Finally, a detailed comparison is made to evaluate the strengths and weaknesses of each architecture, focusing on accuracy, scalability, and practical applicability to sweet orange leaf disease detection.

1.5 Project Outcome

This research successfully develops and evaluates deep learning models for sweet orange leaf disease detection, achieving the highest performance with the hybrid CNN-ViT model. The results show that the hybrid model outperforms both the ViT and CNN models in terms of accuracy and consistency, making it the most effective solution.

The preprocessing techniques applied, such as augmentation and contrast enhancement, played a key role in improving dataset quality and model performance. The standalone CNN and ViT models demonstrated satisfactory results, but their limitations in handling complex disease patterns were evident. The hybrid CNN-ViT model addresses these challenges by combining the local feature extraction capabilities of CNNs with the global feature representation of ViTs, achieving superior performance.

This research highlights the effectiveness of hybrid architectures for agricultural disease detection and provides insights into the comparative performance of ViT, CNN, and hybrid models. The findings contribute to the advancement of sweet orange leaf disease detection and demonstrate the potential for deploying efficient, accurate, and scalable AI solutions in real-world agricultural applications.

1.6 Organization of the Report

This report is systematically organized into six chapters, each addressing critical aspects of the research on sweet orange leaf disease detection using Vision Transformers (ViT), Convolutional Neural Networks (CNN), and a hybrid CNN-ViT model. The structure and content of the report are as follows:

- Chapter 1: Introduction

This chapter provides a comprehensive introduction to the research topic, articulating the motivation, objectives, and methodology of the study. It also outlines the expected outcomes, establishing the context and significance of the work.

- Chapter 2: Background

This chapter presents a detailed review of the existing literature, including similar applications and related research studies. It identifies key gaps in the current body of knowledge, justifying the need for the proposed research.

- Chapter 3: Research Methodology

This chapter elaborates on the methodology adopted for the study, including the requirement analysis, system design, and functional and non-functional specifications. It provides a clear framework of the proposed approach with supporting diagrams, such as the context diagram and data flow diagram, and outlines the project plan and task allocation.

- Chapter 4: Implementation and Results

This chapter focuses on the technical aspects of the research, including the experimental setup, model implementation, and evaluation procedures. It presents the results of the study, accompanied by a critical discussion of the findings and a comparative analysis of the models.

- Chapter 5: Engineering Standards and Design Challenges

This chapter addresses the compliance of the research with relevant software, hardware, and communication standards. It also examines the societal and environmental impacts of the proposed solution, ethical considerations, sustainability aspects, and the challenges encountered during complex engineering problem-solving.

- Chapter 6: Conclusion

The concluding chapter synthesizes the key findings of the research, reflecting on

its contributions and limitations. It offers insights into potential future directions to extend the scope and impact of the study.

The report is structured to provide a logical progression of ideas, from the foundational aspects of the research to the practical outcomes and broader implications. This organization ensures a comprehensive understanding of the study and its significance within the field of agricultural disease detection.

Chapter 2

Background

The detection and classification of plant diseases using deep learning have revolutionized precision agriculture, offering scalable, efficient, and accurate solutions for disease management. The rapid advancement in computer vision and machine learning has facilitated the development of various models for identifying diseases in citrus fruits and leaves. This chapter provides a detailed literature review of the existing methods, focusing on similar applications and related research in plant disease detection using CNNs, Vision Transformers (ViTs), and hybrid models.

2.1 Introduction

The detection and classification of plant diseases using deep learning have revolutionized precision agriculture, offering scalable, efficient, and accurate solutions for disease management. As plant diseases remain a critical challenge for global agriculture, affecting crop yields, economic stability, and food security, the need for automated systems to address these issues has become increasingly important.

The rapid advancements in computer vision and machine learning have facilitated the development of various models capable of identifying diseases in citrus fruits and leaves with high accuracy. Convolutional Neural Networks (CNNs) have traditionally been at the forefront of these efforts, excelling in extracting and classifying localized features. However, the emergence of Vision Transformers (ViTs), which excel at capturing global contextual information, has introduced a new paradigm for plant disease detection.

Hybrid approaches that combine CNNs and ViTs have also gained traction, leveraging the strengths of both architectures to achieve enhanced performance. These methods are particularly relevant for addressing challenges such as dataset variability, scalability, and real-world adaptability. This chapter provides a detailed review of the existing literature, categorizing studies into similar applications—focused specifically

on citrus fruits and leaves—and related research that extends to broader advancements in plant disease detection using CNNs, ViTs, and hybrid models. By analyzing these studies, the chapter aims to establish the context and identify the gaps addressed by this research.

2.2 Literature Review

The advancement of computer vision and deep learning has significantly enhanced the field of plant disease detection and classification, with various models being developed to address the challenges in citrus fruit and leaf disease identification.

Khattak [1] introduced a CNN-based model tailored for distinguishing between healthy and diseased citrus fruits and leaves, achieving a notable test accuracy of 94.55%. Their work emphasized the practicality of such models as decision-support tools for farmers, demonstrating the applicability of deep learning in agricultural contexts. Building upon this foundation, Lanjewar and Parab [2] leveraged transfer learning in combination with deep CNN architectures to classify citrus leaf diseases. Their approach achieved an impressive accuracy of 98% and an F1 score of 99%, highlighting the critical role of image augmentation and the ease of deploying models on cloud platforms for wider accessibility.

Exploring hybrid methodologies, Garg [12] proposed an SVM-CNN model for detecting disorders in oranges, achieving an accuracy of 88.13734%. Their sensitivity analysis underscored the importance of lesion characteristics, showcasing the potential of hybrid approaches in capturing disease-specific features. Similarly, Gupta [9] demonstrated the efficacy of another hybrid CNN-SVM model, achieving an accuracy of 89.6% in lemon leaf disease detection. Their approach focused on the early detection of diseases, which is crucial for timely interventions in agricultural practices.

Further innovations were introduced by Momeny [13], who implemented a learning-to-augment strategy with Bayesian optimization, enhancing the robustness of their CNN model. This approach achieved remarkable results, including a 99.5% accuracy and an F-measure of 100% for identifying black spot disease in oranges. Meanwhile, Pourdarbani [14] extended the application of deep learning to quality control, utilizing 3D-CNN models to classify bruised lemons. Among the tested models, ResNet

demonstrated the highest accuracy of 90.47%, providing valuable insights into the role of 3D imaging in quality assessment.

In the domain of transformers, significant advancements have been achieved. Dümen [4] conducted a comparative study employing eight deep learning methods and two transformer-based approaches for lemon quality classification. Their use of ViT models yielded exceptional accuracy (99.84%), recall (99.95%), and precision (99.66%), establishing transformers as a robust alternative to traditional CNNs. Thai [15] also highlighted the superiority of ViT over CNNs in leaf disease analysis, achieving competitive accuracies. Their work, which involved model quantization for deployment on a Raspberry Pi 4, demonstrated the potential of ViT for resource-constrained applications, paving the way for innovative solutions in smart agriculture.
Expanding the application of ViT to advanced imaging techniques, De Silva and Brown [11] integrated multispectral imaging with ViT models. Their research achieved training and test accuracies of 93.71% and 90.02%, respectively, validating the utility of multispectral imaging under practical field conditions for plant disease detection.

Lee [16] further contributed to the field by implementing an EfficientNet-b0 model within a web application for citrus pest detection, achieving average accuracies and F1 scores of 97% and 96%, respectively. Their study underscores the potential of automated tools to enhance citrus fruit quality and improve agricultural outcomes.

Boukabouya [22] corroborated the efficacy of ViT models in early disease detection, achieving accuracies between 96.7% and 99.7% for tomato leaf disease classification. Their findings underscore the pivotal role of ViT models in promoting sustainable agricultural practices. Wu [17] advanced this concept with a multi-granularity feature extraction model based on ViT, achieving a 2% improvement in accuracy compared to other models. The compact parameter size of their model emphasized its computational efficiency, making it ideal for practical agricultural applications.

Yong [6] emphasized the importance of data quality in disease detection, addressing the scarcity of leaf disease datasets by applying contrast boosting, sharpening, and segmentation techniques. Their hybrid ViT-CNN model demonstrated high compatibility for plantation use, validated through performance metrics and visualization graphs. Similarly, Sudha and Vignesh [18] explored quantum machine

learning approaches for tomato leaf disease detection, where their Quantum-Classical Hybrid Convolution Neural Network (QCHCNN) achieved an impressive accuracy of 99%. Their work highlighted the potential of quantum optimization methods in agricultural disease management.

Fahim-Ul-Islam [19] addressed the challenges of wheat disease classification through federated learning (FL), integrating cutting-edge ViT models such as CoAtNets and Swin Transformer V2. Their pruned architecture reduced computational overhead while maintaining superior accuracy (up to 99%), precision (98%), and recall (98%), demonstrating the effectiveness of distributed learning strategies in agricultural systems. Prashanthi [8] presented LEViT, a ViT-based model enhanced with explainable artificial intelligence (XAI) features, achieving a test accuracy of 92.33% on a diverse dataset of leaf diseases. Grad-CAM integration improved interpretability, enabling users to identify regions influencing model decisions.

Thakur [5] introduced PlantViT, a hybrid ViT-CNN model optimized for plant disease detection. Evaluated on PlantVillage and Embrapa datasets, their model achieved accuracies of 98.61% and 87.87%, respectively, demonstrating significant improvements over state-of-the-art methods. Emon [10] focused on citrus leaf disease detection, employing an ensemble approach that combined segmentation and classification to achieve a prediction accuracy of 95%. Their method highlighted the potential to minimize production losses through timely disease identification.

In the context of agricultural product quality, Dümen [4] employed ViT for lemon classification, achieving an unprecedented accuracy of 99.84%. Their results emphasized the transformative role of ViT in assessing agricultural quality standards. Complementing this, Sanyal [20] proposed a CNN-based grading system for lemons, showcasing the model's ability to streamline the quality assessment process with objectivity and efficiency. Lastly, Pramanik [21] applied transfer learning-based deep learning models for lemon leaf disease classification. Among the models tested, Xception achieved the highest accuracy of 94.34%, highlighting its effectiveness in cost-efficient field-level disease classification.

Table 2.1: Summary of Literature Reviewed.

Author(s) and Year	Model Used	Accuracy	Dataset	Contribution
Khattak et al. (2021)	CNN	94.55%	Citrus fruits and leaves	Developed a CNN model for classifying healthy and diseased citrus fruits and leaves.
Lanjewar and Parab (2023)	Transfer Learning CNN	98%	Citrus leaf images	Leveraged transfer learning and image augmentation for citrus disease classification.
Garg et al. (2023)	SVM-CNN	88.14%	Orange disorders dataset	Proposed a hybrid SVM-CNN model emphasizing lesion characteristics.
Gupta et al. (2023)	CNN-SVM	89.6%	Lemon leaf disease dataset	Developed a hybrid CNN-SVM model for early disease detection.
Momeny et al. (2022)	Bayesian-optimized CNN	99.5%	Black spot disease in oranges	Implemented a learning-to-augment strategy for robust CNN model training.
Pourdarbani et al. (2023)	3D-CNN (ResNet)	90.47%	Bruised lemons	Applied 3D imaging for lemon quality classification.
Lee et al. (2022)	EfficientNet-b0	97%	Citrus pest images	Integrated citrus pest detection into a web application.
Dümen et al. (2023)	ViT	99.84%	Lemon quality classification	Demonstrated the transformative role of ViT for agricultural quality standards.
De Silva and Brown (2023)	ViT + Multispectral Imaging	93.71% (Train), 90.02% (Test)	Multispectral imaging dataset	Integrated multispectral imaging with ViT for practical field conditions.
Boukabouya et al. (2022)	ViT	96.7%-99.7%	Tomato leaf disease dataset	Validated the use of ViT for early tomato leaf disease detection.

Wu et al. (2021)	Multi-granularity ViT	+2% over others	Generic agricultural dataset	Enhanced accuracy with efficient feature extraction for practical use.
Sudha and Vignesh (2024)	Quantum-Classical Hybrid CNN	99%	Tomato leaf disease dataset	Demonstrated the potential of quantum methods in agricultural disease detection.
Fahim-Ul-Islam et al. (2024)	Federated Learning ViT	99%	Wheat leaf disease dataset	Developed pruned ViT for federated learning in resource-constrained settings.
Prashanthi et al. (2024)	LEViT (ViT with XAI)	92.33%	Leaf disease dataset	Enhanced interpretability with Grad-CAM and robust ViT-based model.
Thakur et al. (2021)	PlantViT (Hybrid ViT-CNN)	98.61%	PlantVillage	Optimized ViT-CNN hybrid model for general plant disease detection.
Emon et al. (2023)	Ensemble Model	95%	Citrus leaf disease dataset	Combined segmentation and classification for better predictions.
Pramanik et al. (2021)	Xception	94.34%	Lemon leaf dataset	Applied transfer learning for efficient lemon leaf disease classification.
Pourdarbani et al. (2023)	3D-CNN (ResNet)	90.47%	Bruised lemons	Applied 3D imaging for lemon quality classification.

2.2.1 Similar Applications

Many researchers have worked on developing models to detect diseases in citrus fruits and leaves, which are closely related to this research. For instance, Khattak et al. (2021) introduced a CNN-based model to distinguish between healthy and diseased citrus fruits and leaves, achieving an accuracy of 94.55%. This model demonstrated how deep learning could help farmers make better decisions in managing crop health. Similarly, Lanjewar and Parab (2023) used transfer learning with CNNs to classify citrus leaf diseases, reaching an accuracy of 98%. Their work showed how image augmentation techniques could improve model accuracy.

Some researchers have combined different methods to create hybrid models. Garg et al. (2023) proposed an SVM-CNN model for identifying disorders in oranges, achieving an accuracy of 88.14%. Another study by Gupta et al. (2023) used a CNN-SVM hybrid model for detecting lemon leaf diseases, with an accuracy of 89.6%. These studies highlighted the importance of combining methods to capture disease-specific features.

Beyond disease detection, related applications have also focused on fruit quality and pest detection. Pourdarbani et al. (2023) used a 3D-CNN model with ResNet to classify bruised lemons, achieving 90.47% accuracy. Lee et al. (2022) developed a web application using EfficientNet-b0 for citrus pest detection, achieving an accuracy of 97%. These applications show how deep learning can also improve agricultural product quality and pest management through accessible tools like mobile or web-based platforms.

2.2.2 Related Research

In addition to citrus disease detection, broader research on plant disease detection has introduced advanced methods that can inspire this study. Boukabouya et al. (2022) demonstrated the effectiveness of Vision Transformers (ViTs) for detecting diseases in tomato leaves, achieving accuracies between 96.7% and 99.7%. Their findings showed that ViTs could handle complex datasets better than traditional CNNs.

Hybrid models have also been explored in other crops. Thakur et al. (2021) developed a hybrid ViT-CNN model called PlantViT, which achieved 98.61% accuracy on the PlantVillage dataset. Similarly, Yong et al. (2024) applied preprocessing techniques like contrast enhancement and segmentation to improve their hybrid model's performance, demonstrating its use for plantation-scale disease detection.

Other studies focused on improving computational efficiency and adapting models for resource-limited environments. Wu et al. (2021) used a multi-granularity ViT model, which improved accuracy by 2% compared to traditional models. Thai et al. (2021) optimized ViTs for deployment on low-cost devices like Raspberry Pi, highlighting their potential for real-world applications in small farms.

Emerging technologies have also been applied. Sudha and Vignesh (2024) introduced a quantum-classical hybrid CNN for detecting tomato diseases, achieving 99% accuracy. Fahim-Ul-Islam et al. (2024) used federated learning with ViTs to classify wheat leaf diseases, reaching an accuracy of 99%. These methods show how advanced technologies can make agricultural models faster, more accurate, and accessible.

These studies provide valuable insights into how CNNs, ViTs, and hybrid models can address challenges like data quality, scalability, and computational efficiency in plant disease detection.

2.3 Gap Analysis

The following table identifies gaps and contributions from related authors in the context of features such as technologies, preprocessing, dataset quality, scalability, and hybrid models, aligned with the proposed system.

Table 2.2: Gap Analysis from literature review

Features	Lanjewar and Parab (2023)	Boukabou ya et al. (2022)	Thakur et al. (2021)	Yong et al. (2024)	Proposed System
Technologies (CNN, ViT, Hybrid)	CNN	ViT	Hybrid ViT-CNN	Hybrid ViT-CNN	Hybrid ViT-CNN
Preprocessing Techniques	Image Augmentation	No	Contrast Boosting	Segmentation	Advanced Augmentation
Dataset Quality	Moderate Quality	Tomato Dataset	PlantVillage	Plantation Dataset	Balanced Dataset
Scalability for Larger Classes	No	Yes	Yes	Yes	Yes
Focus on Citrus Diseases	Yes	No	No	Yes	Yes

Integration of Hybrid Models	No	No	Yes	Yes	Yes
Computational Efficiency	Moderate	Efficient	Moderate	Efficient	Optimized

2.4 Summary

This section reviewed the related literature and identified gaps in existing research methodologies and technologies for plant disease detection. The literature highlights the advancements in CNN, ViT, and hybrid models for agricultural applications, with significant contributions in preprocessing techniques, dataset quality, scalability, and hybrid integration. A comparative analysis of related studies revealed the limitations in addressing citrus-specific diseases, interpretability, and computational efficiency, which this research aims to address through the proposed hybrid CNN-ViT system. The gap analysis table further aligned these gaps with the features of the proposed system, establishing the foundation for the methodology and implementation in subsequent chapters.

Chapter 3

Research Methodology

This chapter outlines the systematic approach adopted to design, develop, and evaluate the hybrid deep learning model for sweet orange disease detection. It explains the methodology for collecting and preparing data, implementing the CNN, ViT, and hybrid CNN-ViT models, and evaluating their performance. Furthermore, the chapter details the experimental setup, preprocessing techniques, and evaluation metrics used, ensuring reproducibility and transparency in the research process.

3.1 Methodology/Requirement Analysis & Design Specification

3.1.1 Overview

This section provides a comprehensive outline of the methodology used to develop an advanced system for detecting and classifying sweet orange leaf diseases. It encompasses requirement analysis, design specifications, and the implementation of state-of-the-art deep learning models, including CNN, ViT, and hybrid CNN-ViT architectures. The methodology emphasizes optimizing preprocessing techniques, model training, and evaluation to achieve superior accuracy and reliability. Additionally, scalability and user accessibility are prioritized to ensure the system's practical applicability in real-world agricultural environments.

The research employs Google Colab as the computational platform, leveraging its cloud-based resources to efficiently handle model training and experimentation. The datasets undergo preprocessing steps like augmentation and resizing to enhance data quality and diversity. Finally, performance metrics such as accuracy and F1-score are used to validate the system's effectiveness.

3.1.2 Proposed Methodology/ System Design

The proposed methodology for detecting sweet orange leaf diseases involves a systematic approach to leverage advanced deep learning models for accurate classification. The process begins with dataset preparation, where diverse images of sweet orange leaves, including both healthy and diseased samples, are collected from reliable sources. These images are then subjected to pre-processing steps, such as adjusting DPI for resolution uniformity, resizing to ensure compatibility with deep learning model requirements, and normalizing pixel values to improve computational efficiency and model performance. The pre-processed dataset is used to train three models: a Convolutional Neural Network (CNN) for extracting localized features, a Vision Transformer (ViT) for capturing global contextual information, and a hybrid CNN-ViT model that combines the strengths of both architectures. Each model undergoes rigorous training and evaluation using performance metrics like accuracy and F1-score. Among the models, the hybrid CNN-ViT demonstrates superior performance and is prioritized for disease classification. The system outputs the classification results, which are visualized for interpretability, ensuring usability for agricultural professionals and farmers. This methodology prioritizes scalability, accuracy, and practical applicability, making it a reliable solution for real-world agricultural challenges.

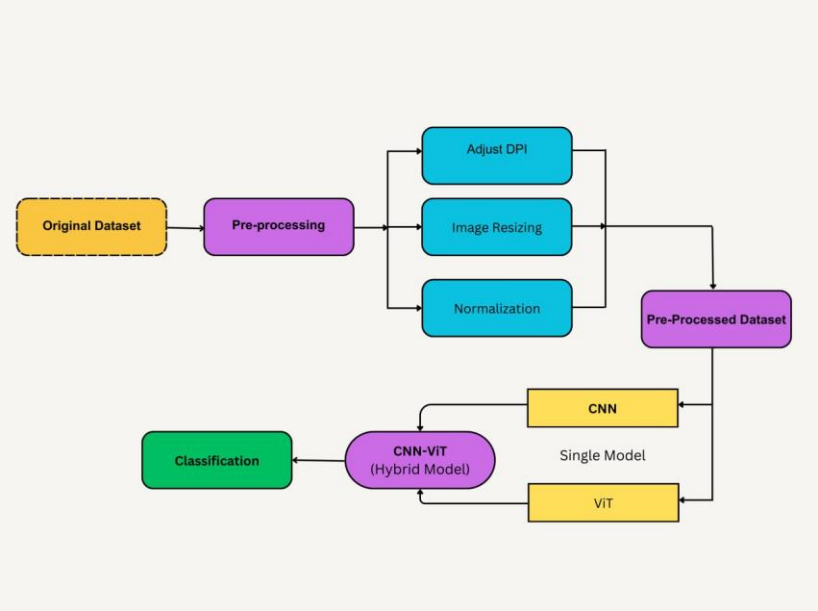

Figure 3.1: The methodological approach of the process.

3.1.3 Functional and Nonfunctional Requirements

Functional Requirements:

- Disease Detection and Classification: The system must accurately detect and classify sweet orange leaf diseases into predefined categories using CNN, ViT, and hybrid CNN-ViT models.

- Data Preprocessing: The system should preprocess input images by applying techniques such as resizing, contrast enhancement, and segmentation to ensure robust model performance.

- Model Evaluation: The platform must evaluate the performance of all models based on metrics such as accuracy, F1-score, precision, and recall.

- Data Integration: The system must support integration with Google Drive for seamless access to datasets and model checkpoints.

- Hybrid Model Implementation: The system must implement and optimize a hybrid CNN-ViT model for enhanced performance and scalability.

- Result Visualization: The system should generate visual outputs such as confusion matrices and accuracy plots to facilitate result interpretation.

- Resource Optimization: The platform should dynamically allocate computational resources (CPU/GPU) based on the user's Google Colab tier.

Nonfunctional Requirements:

- Scalability: The system must be scalable to handle large datasets and adapt to additional crop disease classifications in the future.

- Efficiency: The models should achieve high computational efficiency to ensure compatibility with resource-constrained environments, such as mobile devices or IoT systems.

- Usability: The platform should offer an intuitive interface, making it accessible for

agricultural professionals and non-technical users.

- Reliability: The system must ensure reliable performance under varying environmental conditions and dataset characteristics.

- Security: Data and models stored on Google Drive must remain secure, with access restricted to authorized users.

- Maintainability: The codebase and models should be modular and well-documented for easy maintenance and updates.

- Compliance: The platform must adhere to software standards and ethical considerations, ensuring responsible AI usage in agriculture.

This section defines the key operational and technical requirements to ensure the system meets its objectives effectively and efficiently.

3.2 Detailed Methodology and Design

3.2.1 Convolutional Neural Network (CNN)

In this study, the CNN model provided a strong baseline by accurately classifying disease types based on distinct local features. While its performance was slightly lower than the Vision Transformer (ViT) model, CNN's ability to capture fine-grained patterns led to meaningful insights into how localized disease symptoms appear in sweet orange leaves. This model established a foundation that highlighted the need for additional contextual analysis, later addressed by the hybrid approach.

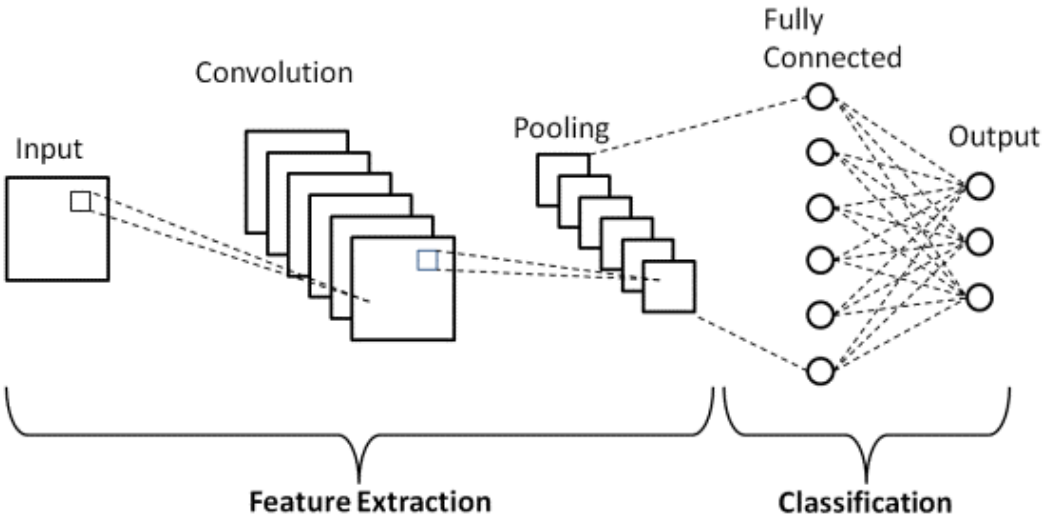

Figure 3.2: The Architecture of Convolutional Neural Network (CNN)

Input Layer - The workflow commences with the input layer, which takes an image of size 224 x 224 x 3. Here, 224 x 224 specifies the image's height and width, while 3 indicates the three-color channels (Red, Green, and Blue). This layer is essential for feeding the raw pixel data into the CNN for processing.

Convolutional Layer 1- In this layer, 64 convolutional filters of size 3 x 3 are applied to the input image. Each filter slides over the image, performing element-wise multiplications and summing the results to create a feature map. This operation extracts low-level features such as edges and textures, resulting in an output feature map of size 224 x 224 x 64. The use of multiple filters allows the network to capture various features simultaneously.

Activation Function (ReLU)- Following the convolution operation, a Rectified Linear Unit (ReLU) activation function is applied. ReLU introduces non-linearity into the model by transforming negative values to zero while retaining positive values. This step is crucial because it enables the network to learn complex patterns and relationships in the data, enhancing its ability to represent intricate features.

Max Pooling Layer 1- A max pooling operation is then performed with a 2 x 2 filter, which reduces the spatial dimensions of the feature map by taking the maximum value from each 2 x 2 block of pixels. This downsampling operation results in a new feature map of size 112 x 112 x 64. Max pooling helps to reduce the computational load and mitigate overfitting by providing a form of translation invariance, meaning the model becomes less sensitive to small translations in the input image.

Convolutional Layer 2- The next step involves applying 128 convolutional filters of size 3 x 3 to the output from the first pooling layer. This layer further refines the feature representation, producing an output feature map of size 112 x 112 x 128. The increased number of filters enables the model to capture more complex patterns and features at this

stage.

ReLU Activation- Another ReLU activation function is applied after the second convolution, introducing non-linearity and allowing the model to learn richer representations from the newly generated feature maps.

Max Pooling Layer 2- A second max pooling operation is performed, reducing the output dimensions to 56 x 56 x 128. This step continues to down-sample the feature map, retaining only the most salient features while reducing computational complexity.

Convolutional Layer 3- This layer applies to 256 filters of size 3 x 3, yielding an output of 56 x 56 x 256. This additional convolutional layer captures even higher-level features, allowing the network to learn more abstract representations of the data.

Max Pooling Layer 3- A third max pooling operation is applied, further downsampling the feature map to 28 x 28 x 256. This process not only reduces the dimensions but also emphasizes the most critical features for the subsequent layers.

Convolutional Layer 4- In this layer, 512 filters of size 3 x 3 are applied to the output, producing an output of 28 x 28 x 512. The increased filter count allows the model to capture an even broader array of features, enhancing the richness of the learned representations.

Max Pooling Layer 4- This layer performs max pooling to down-sample the feature map to 14 x 14 x 512, continuing the trend of reducing dimensions while preserving significant features.

Convolutional Layer 5- In the fifth convolutional layer, 512 filters of size 3 x 3 are again applied, resulting in an output of 14 x 14 x 512. This layer captures even more intricate patterns and features within the data.

ReLU Activation- Following this convolution, the ReLU activation function is applied to allow for complex feature learning.

Max Pooling Layer 5- The final max pooling operation reduces the dimensions to 7 x 7 x 512, significantly condensing the feature representation while retaining the most critical information.

Flattening Layer- After the last pooling layer, the multi-dimensional output is flattened into a one-dimensional vector of size 1 x 1 x 512. This step is crucial for preparing the data for the fully connected layers, transforming the spatial representation into a format suitable for classification.

Fully Connected Layer 1- The flattened output is then passed into the first fully connected layer, which consists of 4096 neurons. This layer learns high-level features by combining the information from the flattened input and applies a ReLU activation function. This connection allows for a more comprehensive integration of the learned features across the previous layers.

Fully Connected Layer 2- The output from the first fully connected layer is fed into a second

fully connected layer with 1000 neurons. This layer generates scores for each class in the classification task, effectively assessing how likely the input image belongs to each of the predefined categories.

SoftMax Layer- Finally, a SoftMax activation function is applied to the output of the second fully connected layer. This function converts the raw class scores into probabilities, ensuring that the sum of all output probabilities equals 1. The result is a probability distribution over the classes, which indicates the likelihood of the input image belonging to each class, facilitating the classification task.

Overall, this CNN architecture processes the input image through a series of convolutional layers, activation functions, and pooling operations, progressively extracting and abstracting features at each stage. By the end of the workflow, the network produces a probability distribution that aids in accurately classifying the input image, highlighting the hierarchical learning process that enables CNNs to capture complex patterns and achieve high performance in image classification tasks.

3.2.2 Vision Transformer (ViT)

In this study, the ViT model was applied to the sweet orange disease dataset to leverage its ability to capture broad contextual features. ViT outperformed CNN by offering a broader contextual understanding of disease patterns across entire leaves. This global perspective enabled the model to capture subtle, distributed symptoms that CNN alone might miss. ViT's ability to model relationships across distant image regions provided meaningful insights into the complex demonstration of certain diseases, highlighting the value of global context in agricultural disease detection.

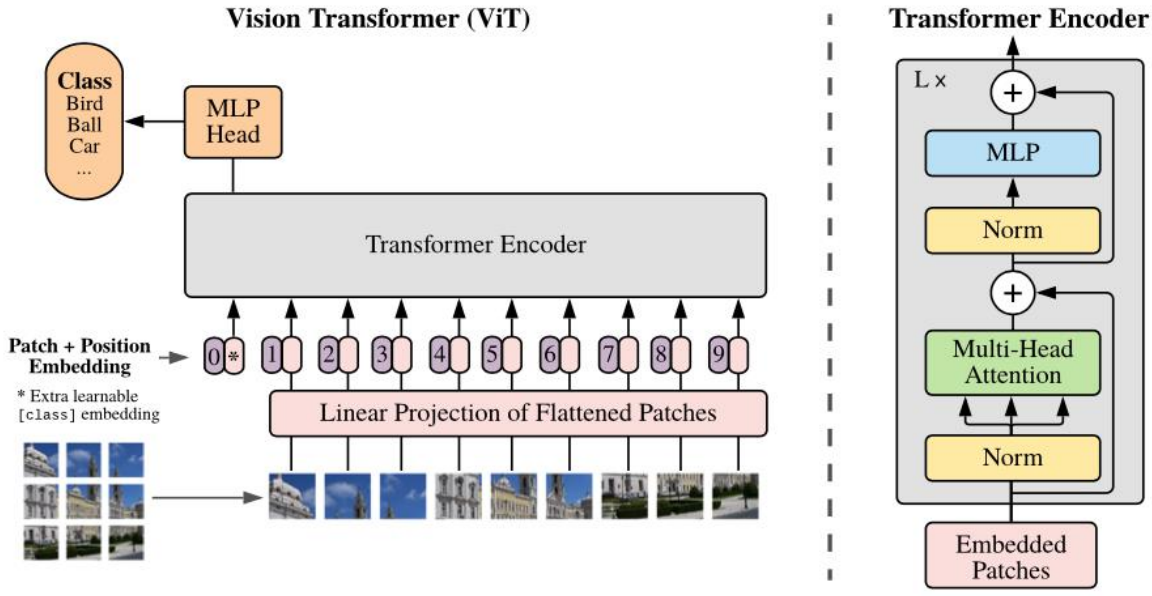

Figure 3.3: The Architecture of Vision Transformer (ViT)

Patch Embedding

Image to Patches- The Vision Transformer starts by dividing the input image into multiple fixed-size patches. In this study, the image size is 240x240 pixels and the patch size is 16x16, the model splits the image into a grid of 14 x 14 patches, resulting in 196 patches in total. Each patch can be thought of as a smaller image, containing local details of the larger image.

Flattening- Each patch is then "flattened" into a single vector by concatenating its pixel values. Suppose each patch has a size of 16x16 pixels with 3 color channels (RGB); flattening would yield a vector of length 768 (16 x 16 x 3). This flattening operation converts the 2D patch into a 1D vector, making it compatible with the subsequent layers.

Linear Projection- After flattening, each patch vector is passed through a linear projection layer. This layer transforms each vector into a fixed-size embedding, typically of a dimension like 768. The linear projection is effectively a learned mapping that encodes each patch's features into a higher-dimensional space, which is more suitable for learning complex patterns.

Result- By the end of this step, we have a sequence of patch embeddings, each representing one portion of the original image. This sequence of embeddings is like the sequence of tokens (words) used in natural language processing (NLP) models.

Adding Position Embeddings

Positional Embeddings- Unlike CNNs, which preserve the spatial structure of images through convolution operations, transformers treat each patch as a standalone "token." This token-based approach means transformers lack an inherent understanding of spatial relationships (e.g., which patches are adjacent). To address this, positional embeddings are added to each patch embedding to give the model a sense of order.

Learned Positional Encoding- Positional encodings are vectors of the same dimension as the patch embeddings, which are learned during training. Each position (e.g., first patch, second patch, etc.) is assigned a unique positional vector, which is added to the corresponding patch embedding.

Class Token- Alongside the patch embeddings, an additional, learnable "[class]" token is added at the start of the sequence. This token acts as a placeholder for the final representation of the entire image after the transformer encoder processes all patches. By the end of the encoding process, this class token will contain aggregated information from all the patches, representing the entire image.

Transformer Encoder

The sequence of patch embeddings, including the class token, is fed into a transformer encoder consisting of multiple identical layers. Each encoder layer is made up of the following components:

Self-Attention Mechanism- The self-attention mechanism allows each patch embedding to "attend" to every other patch in the sequence. In other words, each patch can learn which other patches are relevant for understanding its context. This is especially useful for capturing long-range dependencies, such as edges or textures that span across different patches.

Multi-Headed Approach- The transformer uses multiple "heads" (or separate attention mechanisms) in parallel, which enables the model to learn different relationships simultaneously. For example, one head might focus on color patterns, another on textures, and another on object boundaries.

Weighted Summation- For each patch, the self-attention mechanism calculates "attention scores" that determine how much focus should be placed on other patches. The attention scores are used to compute a weighted sum of the other patches, creating a new representation for each patch that incorporates information from its neighbors.

Layer Normalization

After the multi-head self-attention operation, layer normalization is applied to the outputs. Normalization stabilizes the training process by reducing the internal covariate shift (the variation in input distributions across layers), which helps the model to converge more smoothly.

Residual Connection- Additionally, a residual (or skip) connection is applied, meaning the input to the multi-head attention layer is added back to its output. This preserves information from earlier layers and helps gradients flow through the network during backpropagation.

Feed-Forward Neural Network (MLP)

Each encoder layer also includes a feed-forward neural network, often consisting of two fully connected (dense) layers with a non-linear activation function (such as ReLU) in between. This feed-forward network further processes each patch embedding, allowing the model to learn more complex features. While the self-attention mechanism focuses on learning relationships between patches, the MLP expands the depth of representation for each patch.

Normalization and Residual Connection- Like the self-attention component, the MLP also includes layer normalization and residual connections, ensuring stable training and

preserving information flow across layers.

Stacking Layers- The entire transformer encoder, including the multi-head attention, MLP, and normalization layers, is repeated multiple times (typically 12 or 24 layers) to create a deeper network. This stacking of layers allows the model to learn increasingly abstract and high-level features as the information passes through the layers.

Classification Head (MLP Head)

Extracting the Class Token- After passing through all transformer encoder layers, the output of the [class] token (the special token added at the beginning of the sequence) is extracted. This token has accumulated information from all other patches through the self-attention mechanism, making it a comprehensive representation of the entire image.

Multi-Layer Perceptron (MLP) Head- The class token is then passed into a final MLP head, which serves as the classification layer. This MLP is typically a simple neural network with one or two fully connected layers, converting the class token's representation into the final class scores.

Generating Class Scores- The output from the MLP head is a set of class scores, one for each possible category (e.g., bird, ball, car, etc.). The scores represent the model's confidence in the image belonging to each category.

SoftMax for Probabilities

In many cases, the scores are passed through a SoftMax layer to convert them into probabilities, making it easier to interpret the model's prediction by indicating the likelihood of the image belonging to each class.

In this study, by applying vision transformers, the image is divided into small patches, which are linearly projected and embedded with positional information to retain spatial structure. These patches, along with a special class token, are processed through multiple layers of transformer encoders, each consisting of multi-head self-attention, normalization, and feed-forward networks. By the end of the transformer encoder, the class token has aggregated information from all patches, representing the entire image. This class token is passed through a final MLP head to generate classification scores, identifying the category to which the input image belongs.

3.2.3 The Hybrid ViT-CNN Model

This hybrid approach was applied to classify sweet orange leaf diseases more accurately. By integrating CNN and ViT, the hybrid model achieved superior accuracy and robustness compared to the individual models. It successfully combined local and global insights, resulting in more reliable disease classification and deeper insights into how different

disease symptoms interact across an entire leaf. This model demonstrated the effectiveness of a multi-perspective approach, underscoring the potential of hybrid architectures in enhancing disease classification performance in agriculture.

The workflow of this ViT-CNN hybrid model for sweet orange leaf disease detection begins with data preparation. First, a dataset of labeled sweet orange leaf disease images is loaded and organized. Labels are assigned, and image preprocessing is performed to prepare the data. The data is then split into training, validation, and test sets to enable model training and evaluation.

Next, the CNN and ViT models are initialized. The CNN model is configured with an input shape, convolutional layers using ReLU activation, max pooling, and global average pooling. A dense layer with ReLU activation and a dropout rate of 50% is added to prevent overfitting.

Following this, the ViT model is configured by taking the output from the CNN model. The ViT model includes a dense layer, multi-head self-attention with 8 heads, and a feed-forward layer using GELU activation, global average pooling, and dropout (set at 20%). A final dense layer with SoftMax activation provides the output, which represents 8 classes of disease types.

The hybrid ViT-CNN model is then created by combining the CNN and ViT components. Once combined, hyperparameter optimization is performed by selecting image size, batch size, and the number of training epochs. The model is then trained on the training dataset, with evaluation metrics such as accuracy and loss recorded for each configuration.

Finally, model evaluation and export are conducted. The trained model is evaluated using the test dataset to assess its performance, and the best-performing model is saved as integrated_model.h5. This model file represents the optimal sweet orange leaf disease detection system, ready for deployment or further analysis.

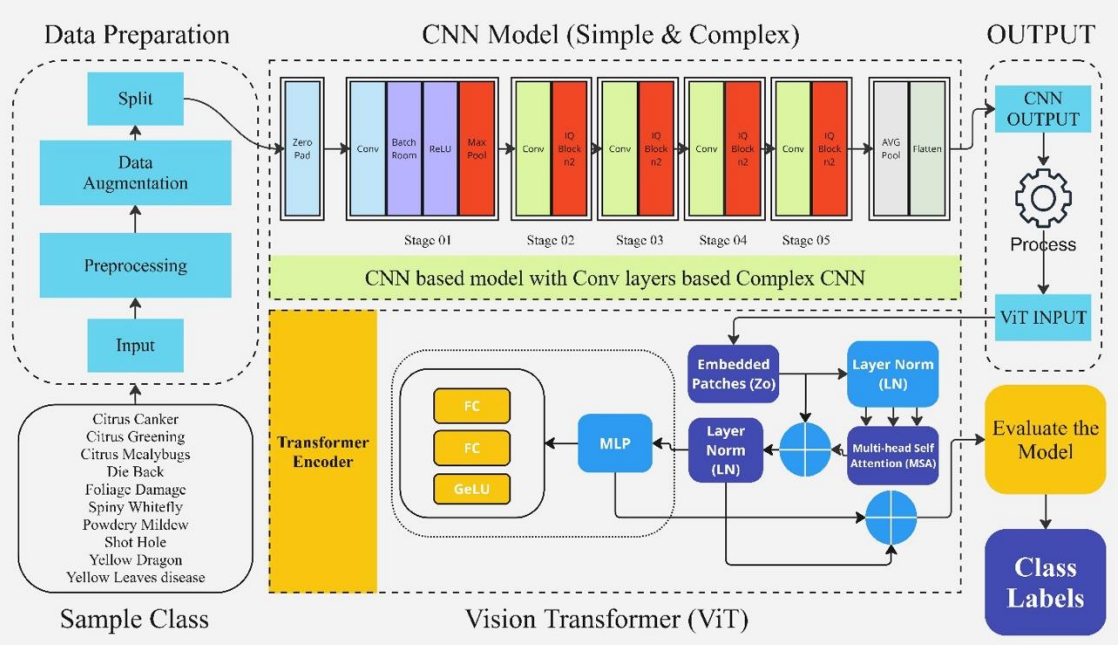

Figure 3.4: Architecture of Hybrid ViT and CNN

3.3 Project Plan

Table 3.1: GANTT Chart of Project Timeline.

Process	May'24	June'24	July'24	Aug'24	Sep'24	Oct'24	Nov'24	Dec'24
Working Plan								
Theoretical Study								
Literature Review								
Data Collection								
Data Preprocessing								
Model Design								
Methodology Writing								
Report Writing								
Review and Finalization								

3.4 Task Allocation

Table 3.2: Task allocation of the project

Task	Details	Duration
Introduction	Defined the problem statement, research objectives, and significance of the study.	Week 1 - Week 2
Background	Conducted an extensive literature review and identified research gaps.	Week 3 - Week 5

Research Methodology	Designed the methodology, including data collection, preprocessing techniques, and model selection.	Week 6 - Week 8
Implementation and Results	Implemented deep learning models (ViT, Mobile ViT, etc.), conducted experiments, and analyzed results.	Week 9 - Week 12
Engineering Standards and Design Challenges	Addressed ethical, societal, and sustainability considerations and overcame technical challenges.	Week 13
Conclusion	Summarized findings, contributions, and outlined future research directions.	Week 14

3.5 Summary

The methodology chapter outlines the systematic approach adopted for developing an efficient system to detect and classify sweet orange leaf diseases. The chapter begins with an overview of the methodology, detailing the requirement analysis and design specifications, followed by a clear explanation of the proposed system design. The proposed methodology includes the collection and preparation of datasets, where images undergo essential pre-processing steps such as resolution adjustment, resizing, and normalization to ensure compatibility with deep learning models.

Three types of models—Convolutional Neural Networks (CNNs), Vision Transformers (ViTs), and a hybrid CNN-ViT—are implemented to classify leaf diseases. These models are trained, validated, and evaluated using performance metrics like accuracy and F1-score. Among these, the hybrid CNN-ViT model demonstrates superior performance, combining the localized feature extraction of CNNs and the global contextual understanding of ViTs.

The chapter emphasizes the importance of scalability, accuracy, and real-world usability. The use of Google Colaboratory for cloud-based implementation ensures efficient computational handling, while the integration of preprocessing and lightweight models enhances the system's accessibility for resource-constrained environments. This methodology provides a robust and reliable framework for addressing sweet orange leaf disease detection, ensuring practical applicability in agricultural settings.

Chapter 4

Implementation and Results

4.1 Environment Setup

The implementation, training, and evaluation of the hybrid CNN-ViT model for sweet orange disease detection were conducted using Google Colab, a cloud-based platform that provides a GPU-powered environment for deep learning experiments. This setup ensured high computational efficiency and accessibility.

Google Colab Configuration

- Runtime Type: GPU (NVIDIA Tesla T4)
- RAM: 12 GB (High-RAM runtime enabled when required)
- Python Version: Python 3.9

Software and Libraries

Deep Learning Frameworks:

- TensorFlow 2.8.0 for model development and training
- PyTorch 1.11.0 for additional experimentation

Supporting Libraries:

- NumPy for numerical computations
- OpenCV for image preprocessing and augmentation
- Matplotlib for visualizations
- Scikit-learn for performance evaluation metrics (e.g., accuracy, F1-score)
- Argumentation for advanced image augmentation techniques

Dataset Access and Management

The dataset used for this study was stored on Google Drive. The following steps ensured smooth integration:

- Drive Mounting: The Google Drive account was mounted within Colab for seamless access to the dataset.
- Dataset Preprocessing: Images were preprocessed directly within the Colab environment, including resizing, contrast adjustment, and augmentation.

Experimentation Workflow

The experiments were organized in Google Colab using modular notebooks:

- Preprocessing Notebook: Responsible for loading, resizing, and augmenting the dataset.
- Model Training Notebook: Implemented and trained the CNN, ViT, and hybrid CNN-ViT models with periodic checkpointing.
- Evaluation Notebook: Evaluated model performance using various metrics and visualized the results.
- Result Storage: All model checkpoints, logs, and generated results were stored back into Google Drive for further analysis and documentation.

4.2 Testing and Evaluation/Performance/ Comparative Analysis

This section evaluates and compares the performance of the experimented models, namely ViT, ResNet50v2, and the hybrid ViT-CNN, based on their train, validation, and test accuracies. These metrics provide insights into the learning and generalization capabilities of each model.

4.2.1 Performance analysis of experimented models

The table compares the train, validation, and test accuracies of three models: Vision Transformer (ViT), ResNet50v2, and the hybrid Vision Transformer-Convolutional Neural Network (ViT-CNN). These metrics highlight each model's learning ability on the training data, as well as their capacity to generalize to unseen validation and test datasets.

The ViT model achieves a training accuracy of 92.78% and validation accuracy of 90%, but its test accuracy is slightly lower at 89.9%. This suggests that while the model learns reasonably well from the training data, it might be less effective in capturing the complex patterns needed for strong generalization to unseen data. The gap between training and

test accuracy indicates potential underfitting or sensitivity to the dataset.

The ResNet50v2 model, in contrast, demonstrates strong and consistent performance across all metrics. With a training accuracy of 97.92%, validation accuracy of 97%, and test accuracy of 97.92%, the model shows excellent generalization capabilities. The small difference between training and test accuracies reflects the model's ability to effectively learn patterns while avoiding overfitting, making it a reliable choice for this task.

The ViT-CNN hybrid model outperforms both ViT and ResNet50v2, achieving the highest test accuracy of 98.26% while maintaining a strong training accuracy of 96.74% and a validation accuracy of 97%. This result highlights the hybrid model's effectiveness in leveraging the strengths of both Vision Transformers and Convolutional Neural Networks. By combining the attention mechanism of ViT with the spatial feature extraction of CNNs, the ViT-CNN model demonstrates superior performance and adaptability, making it the most suitable approach for the task.

In conclusion, while all three models perform well, the ViT-CNN hybrid achieves the best balance between training, validation, and test performance, showcasing its robustness and potential for practical applications. ResNet50v2 is a close second, with consistent and reliable performance, while ViT, although effective, lags slightly behind in generalization to the test data.

Table 4.1: Result Comparison of three different models

Model	Training Accuracy	Validation Accuracy	Test Accuracy
ViT	0.93	0.90	0.90
ResNet50v2	0.97	0.97	0.97
ViT-CNN	0.97	0.97	0.98

4.3 Results and Discussion

4.3.1 Performance Analysis of Vision Transformer (ViT) Model

The figure illustrates the performance analysis of the Vision Transformer (ViT) model through two metrics: accuracy and loss, evaluated over 60 training epochs. The left plot represents the model's accuracy during the training and validation phases, while the right plot shows the corresponding loss curves. These metrics provide insights into the model's learning progress and generalization capability.

In the accuracy plot, the training accuracy increases rapidly in the initial epochs and stabilizes close to 100%, demonstrating the model's ability to effectively learn from the

training data. The validation accuracy also improves steadily, stabilizing around 90%, which indicates that the model generalizes well to unseen data. The slight gap between training and validation accuracy suggests minimal overfitting, reflecting a balanced learning process.

The loss plot complements this analysis by showing a consistent decline in both training and validation loss values over the epochs. The training loss decreases steadily and converges to a minimal value, while the validation loss follows a similar trend and stabilizes at a low level. This alignment between training and validation loss further confirms that the model achieves convergence without significant overfitting. Overall, these results highlight the ViT model's effectiveness and robustness in learning patterns from the dataset while maintaining good generalization.

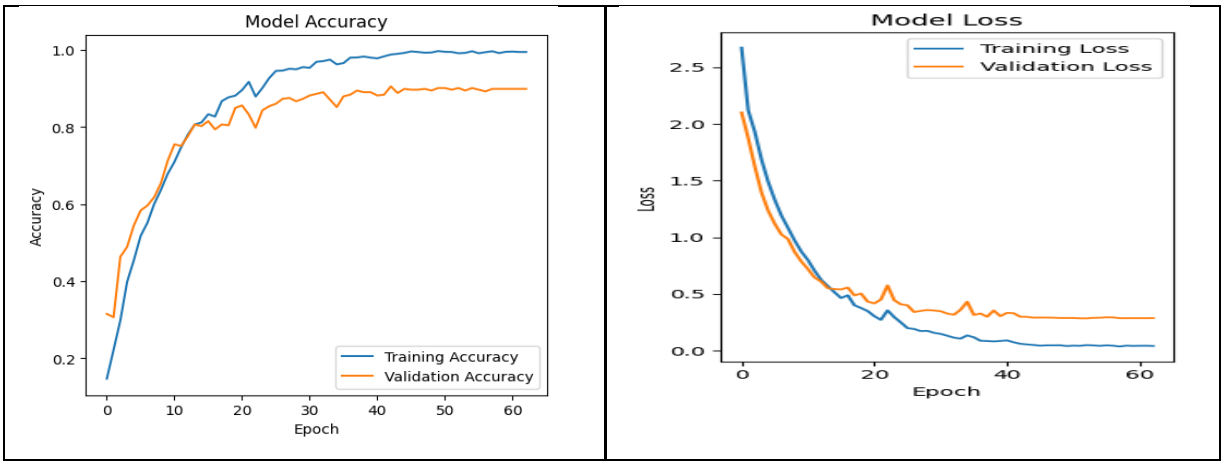

Figure 4.1: Validation Loss and accuracy curve of validation ViT model

The table presents the classification performance of the model on the validation dataset using precision, recall, F1-score, and support for each class. These metrics provide a comprehensive evaluation of how well the model identifies each disease category and healthy leaves in the dataset. Overall, the model achieves a high level of accuracy and consistent performance across most classes.

Precision indicates the proportion of correctly predicted instances out of all predicted instances for a class. Most classes, such as "Foliage Damaged," "Healthy Leaf," and "Shot Hole," exhibit high precision, signifying that false positives are minimal for these categories. Classes like "Yellow Dragon" achieve perfect precision (1.00), ensuring no false positives for this class. However, "Die Back" shows a slightly lower precision (0.79), which could suggest some overlap with other classes in the predictions.

Recall measures the ability of the model to correctly identify all actual instances of a class. Classes like "Foliage Damaged," "Shot Hole," and "Spiny Whitefly" achieve perfect recall

(1.00), demonstrating that the model successfully identifies all true cases of these categories. In contrast, "Yellow Dragon" has a recall of 0.69, indicating that some instances of this class were missed, likely due to underrepresentation in the dataset (only 13 instances in support).

The F1-score, which balances precision and recall, reflects the overall effectiveness of the model for each class. Most classes achieve strong F1-scores, such as "Shot Hole" with 0.99 and "Foliage Damaged" with 0.97, highlighting the model's robust performance. However, categories like "Citrus Canker" (0.80) and "Yellow Dragon" (0.82) exhibit slightly lower F1-scores due to reduced recall in these classes, which may be attributed to fewer samples or greater class complexity.

On a broader scale, the model demonstrates an overall accuracy of 90%, with a macro average F1-score of 0.89 and a weighted average F1-score of 0.90. These averages suggest the model performs well across all classes, with the weighted average reflecting it s ability to handle class imbalances effectively. Despite minor challenges with certain underrepresented classes, the model shows consistent and reliable performance in detecting and classifying diseases within the validation dataset.

Table 4.2: Validation Set classification report

	Precision	Recall	F1 - Score	Support
Citrus Canker	0.86	0.74	0.80	43
Citrus Greening	0.92	0.95	0.93	58
Citrus Mealybugs	0.87	0.96	0.91	27
Die Back	0.79	0.79	0.79	61
Foliage Damaged	0.95	1.00	0.97	55
Healthy Leaf	0.96	0.92	0.94	49
Powdery Mildew	0.84	0.82	0.83	56
Shot Hole	0.99	1.00	0.99	78
Spiny Whitefly	0.87	1.00	0.93	26
Yellow Dragon	1.00	0.69	0.82	13
Accuracy			0.90	466
Macro Avg	0.90	0.89	0.89	466
Weighted Avg	0.90	0.90	0.90	466

The confusion matrix provides a detailed breakdown of the model's classification performance on the validation dataset, showing the actual versus predicted labels for each class. Each row represents the true class, while each column represents the predicted class. The diagonal entries indicate correctly classified instances, while off-diagonal entries

reflect misclassifications.

The majority of the predictions fall on the diagonal, demonstrating the model's strong performance in correctly identifying most instances across all classes. For example, "Shot Hole" has 78 correctly predicted instances, with no misclassifications, showcasing perfect performance for this class. Similarly, "Foliage Damaged" has all 55 instances correctly classified, reflecting the model's high accuracy in identifying this category.

Misclassifications are observed in some classes, though their occurrence is minimal. For instance, in the "Citrus Canker" class, out of 43 actual instances, 32 are correctly classified, while 8 are misclassified as "Die Back" and 2 as "Powdery Mildew." This suggests some overlap in feature representation between these classes. Similarly, "Yellow Dragon," which has only 13 instances in the dataset, shows 4 misclassifications into "Die Back," indicating potential challenges in recognizing this minority class due to limited samples.

Other minor misclassifications include "Powdery Mildew," where 4 instances are predicted as "Die Back," and "Healthy Leaf," where 2 instances are incorrectly labeled. These errors highlight areas for improvement, particularly for classes with fewer samples or overlapping features.

Overall, the confusion matrix demonstrates that the model performs exceptionally well for most classes, with a few misclassifications in minority or visually similar categories. These insights could guide further optimization of the model, such as addressing data imbalance or refining class-specific feature extraction.

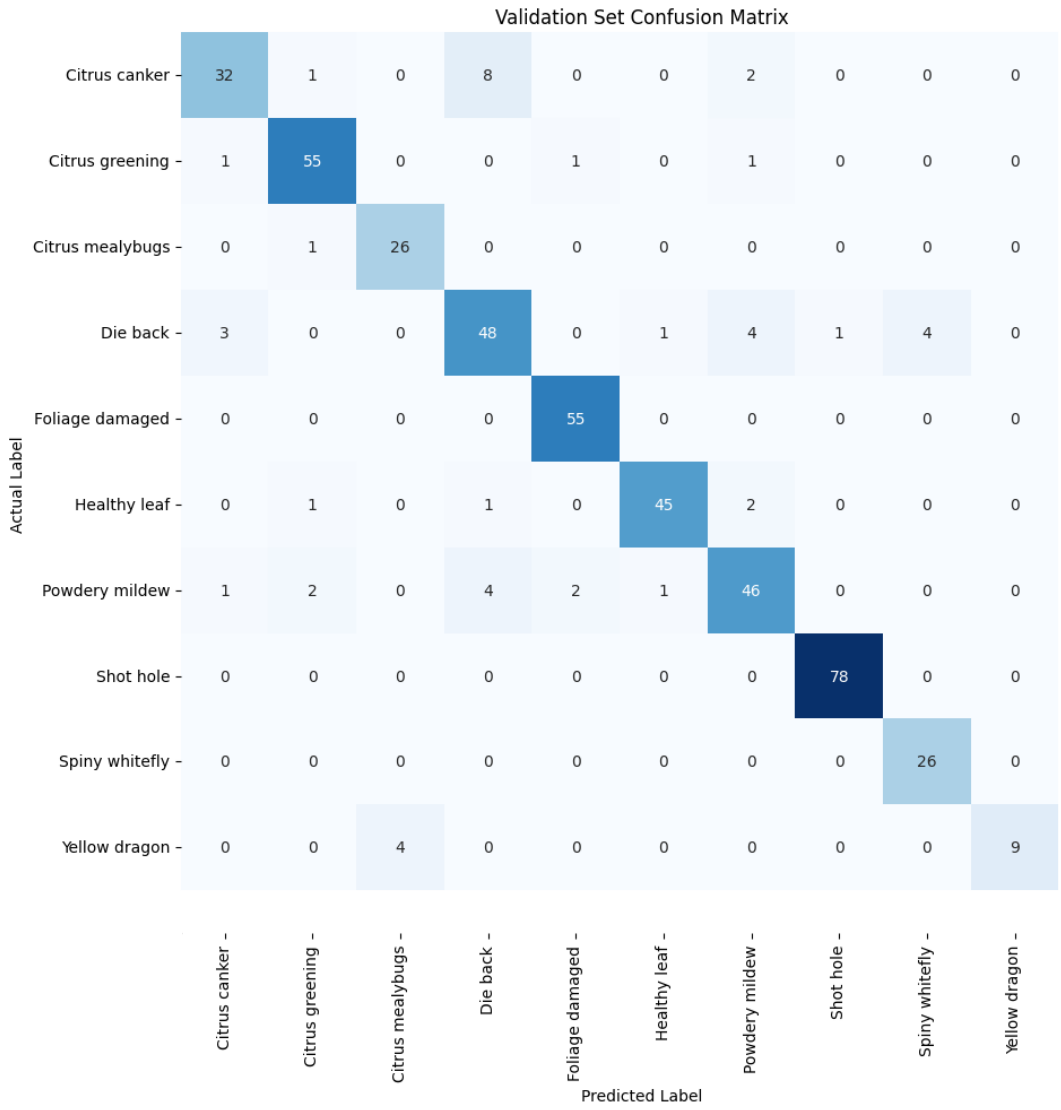

Figure 4.2: Confusion matrix of the validation set

The classification report summarizes the model's performance on the test set using metrics such as precision, recall, F1-score, and support for each class. It provides insights into the model's ability to generalize its predictions to unseen data and highlights areas of strength and opportunities for improvement.

For most classes, the model demonstrates strong performance. For example, "Shot Hole" achieves perfect scores across all metrics, with a precision, recall, and F1-score of 1.00, indicating that the model is highly effective at identifying this class without any errors. Similarly, "Citrus Mealybugs" and "Foliage Damaged" show excellent performance, with F1-scores of 0.97 and perfect recall of 1.00, reflecting the model's ability to correctly identify all instances of these categories. Classes such as "Citrus Greening" and "Healthy Leaf" also perform well, with F1-scores of 0.94 and 0.88, respectively.

However, some classes exhibit moderate performance. For instance, "Die Back" achieves a recall of 0.72 and an F1-score of 0.76, suggesting that the model struggles to correctly

identify all instances of this class. Similarly, "Powdery Mildew" and "Yellow Dragon" show lower recall values of 0.77 and 0.80, respectively, indicating missed predictions. These challenges might stem from overlapping features with other classes or the limited number of samples in the dataset, particularly for "Yellow Dragon," which has only 5 instances. Overall, the model achieves an accuracy of 90% on the test set, indicating strong generalization to unseen data. The macro average F1-score is 0.90, reflecting balanced performance across all classes, while the weighted average F1-score is also 0.90, accounting for class imbalance. These results demonstrate the model's robustness in classifying most disease categories effectively, with some opportunities for further optimization in classes with lower recall or fewer samples.

Table 4.3: Test set classification report

	Precision	Recall	F1 - Score	Support
Citrus Canker	0.79	0.91	0.85	34
Citrus Greening	0.92	0.97	0.94	35
Citrus Mealybugs	0.95	1.00	0.97	18
Die Back	0.81	0.72	0.76	40
Foliage Damaged	0.94	1.00	0.97	34
Healthy Leaf	0.91	0.85	0.88	34
Powdery Mildew	0.86	0.77	0.81	31
Shot Hole	1.00	1.00	1.00	42
Spiny Whitefly	0.93	0.93	0.93	15
Yellow Dragon	1.00	0.80	0.89	5
Accuracy			0.90	288
Macro Avg	0.91	0.90	0.90	288
Weighted Avg	0.90	0.90	0.90	288

The confusion matrix visualizes the classification performance of the model on the test set, detailing the actual versus predicted labels for each class. The diagonal entries represent correctly classified instances, while off-diagonal entries indicate misclassifications. Overall, the matrix demonstrates strong performance, with most predictions aligning correctly with the actual labels.

The model achieves high accuracy for several classes. For example, "Shot Hole" shows perfect classification with all 42 instances correctly predicted and no misclassifications. Similarly, "Citrus Mealybugs" and "Foliage Damaged" are perfectly predicted with no off-diagonal entries, demonstrating the model's reliability in identifying these categories. Other classes, such as "Citrus Greening" and "Healthy Leaf," show predominantly correct predictions with minor misclassifications.

However, some challenges are evident in specific classes. For "Citrus Canker," while 31 out of 34 instances are correctly classified, three instances are misclassified as "Die Back," suggesting potential overlap in the features of these two classes. The "Die Back" class shows more noticeable misclassifications, with 29 correct predictions and several instances misclassified into other classes, including "Powdery Mildew" and "Citrus Canker." This highlights the need for further refinement in distinguishing "Die Back" from visually or feature-similar categories.

"Powdery Mildew" and "Yellow Dragon" also exhibit some misclassifications. For instance, "Powdery Mildew" has 24 correct predictions, but a few instances are misclassified as "Citrus Greening" and "Healthy Leaf." "Yellow Dragon," a minority class with only 5 instances, has 4 correctly predicted samples, but one is misclassified into "Die Back," possibly due to limited training data for this category.

Overall, the confusion matrix highlights the model's strengths in accurately predicting most classes, particularly dominant and visually distinct categories. However, it also reveals opportunities for improvement in minority or overlapping classes, which may require enhanced feature extraction, additional data augmentation, or addressing class imbalances. These insights provide a basis for further optimization to enhance classification performance.

Figure 4.3: Confusion matrix of the test set

4.3.2 Performance Analysis of Convolutional Neural Network (CNN) Model

The figure illustrates the performance analysis of the Convolutional Neural Network (CNN) model using two key metrics: accuracy and loss. The left plot shows the training and validation accuracy over 50 epochs, while the right plot represents the training and validation loss for the same period. These plots provide insights into the model's learning behavior and generalization capability.

In the accuracy plot, the training accuracy quickly increases and stabilizes close to 100%, indicating that the model effectively learns from the training data. The validation accuracy follows a similar trend, steadily improving and stabilizing just below the training accuracy. This slight gap between training and validation accuracy reflects good generalization with minimal overfitting, as the validation accuracy maintains high values throughout the

training process.

The loss plot complements the accuracy analysis by showing the reduction in training and validation loss. The training loss decreases rapidly in the initial epochs and stabilizes at a minimal value, reflecting effective convergence of the model. Similarly, the validation loss decreases steadily and stabilizes at a low value, further confirming the model's ability to generalize well to unseen data.

Overall, the plots indicate that the CNN model achieves high accuracy and demonstrates excellent convergence during training. The minimal gap between training and validation metrics suggests that the model balances learning effectively without overfitting, making it a robust solution for the given dataset.

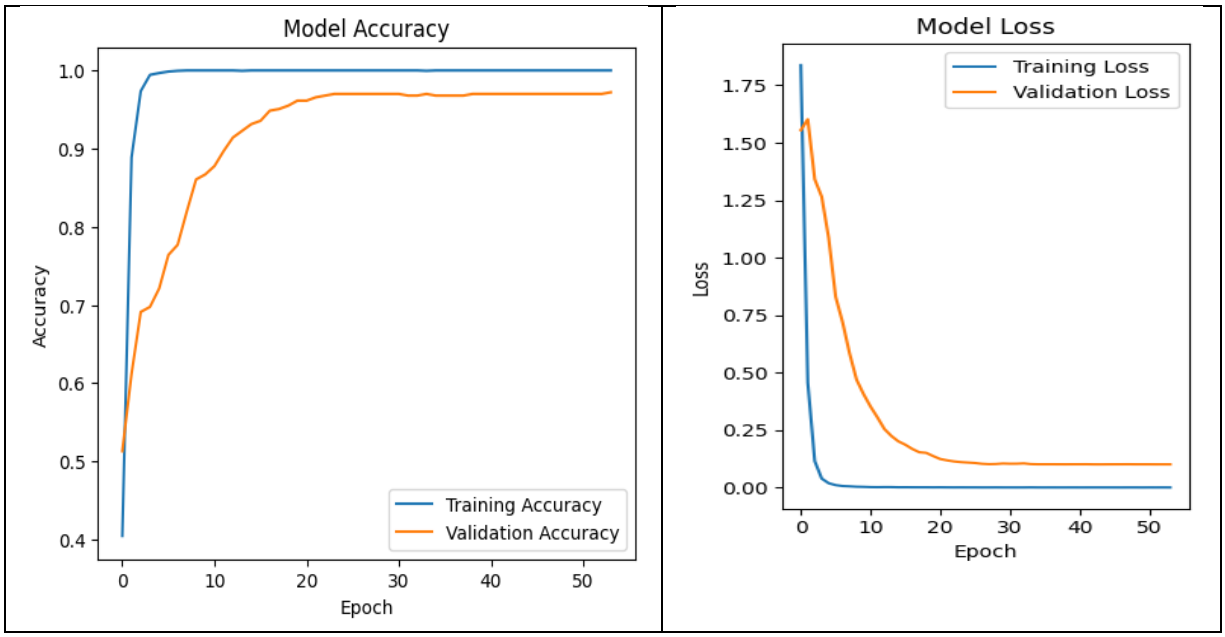

Figure 4.4: Test Loss and accuracy curve of validation CNN model

The table presents the classification performance of the model on the validation dataset, summarizing precision, recall, F1-score, and support for each class. The overall metrics demonstrate the model's strong capability to accurately classify most categories in the dataset.

The model achieves excellent performance for several classes, such as "Foliage Damaged," "Spiny Whitefly," and "Shot Hole," with precision, recall, and F1-scores of 1.00, indicating perfect classification for these categories. Similarly, "Healthy Leaf" and "Die Back" also show high scores, with F1-scores of 0.98 and 0.97, respectively, highlighting the model's ability to accurately detect these classes with minimal misclassifications.

Most other classes, such as "Citrus Greening," "Citrus Mealybugs," and "Powdery Mildew," also exhibit strong performance with F1-scores ranging from 0.95 to 0.97. However, "Citrus

Canker" shows slightly lower recall (0.86), resulting in an F1-score of 0.91, indicating that some true instances of this class were not detected. Similarly, "Yellow Dragon," which has the least support in the dataset (13 instances), achieves perfect precision but lower recall (0.77), leading to an F1-score of 0.87. This suggests challenges in correctly identifying all instances of minority classes due to limited sample size.

The overall accuracy of the model is 97%, indicating that the vast majority of predictions across all classes are correct. The macro average F1-score is 0.96, reflecting balanced performance across all classes, while the weighted average F1-score is 0.97, accounting for class imbalance in the dataset. These metrics collectively highlight the robustness and reliability of the model in classifying most disease and healthy leaf categories with high precision, recall, and F1-scores. Minor areas for improvement are observed in underrepresented classes, which could benefit from additional data or enhanced feature extraction.

Table 4.4: Validation Set classification report

	Precision	Recall	F1 - Score	Support
Citrus Canker	0.97	0.86	0.91	43
Citrus Greening	0.97	0.98	0.97	58
Citrus Mealybugs	0.90	1.00	0.95	27
Die Back	0.97	0.97	0.97	61
Foliage Damaged	1.00	1.00	1.00	55
Healthy Leaf	0.96	1.00	0.98	49
Powdery Mildew	0.96	0.96	0.96	56
Shot Hole	0.97	1.00	0.99	78
Spiny Whitefly	1.00	1.00	1.00	26
Yellow Dragon	1.00	0.77	0.87	13
Accuracy			0.97	466
Macro Avg	0.97	0.95	0.96	466
Weighted Avg	0.97	0.97	0.97	466

The confusion matrix provides a detailed overview of the model's classification performance on the validation set by comparing actual labels with predicted labels for each class. The diagonal values represent correctly classified instances, while the off-diagonal values highlight misclassifications. Overall, the model demonstrates strong performance, as most predictions fall on the diagonal.

Several classes achieve near-perfect classification. For example, "Citrus Mealybugs," "Foliage Damaged," "Healthy Leaf," and "Shot Hole" show no misclassifications, with all

their instances correctly predicted. Similarly, "Spiny Whitefly" and "Yellow Dragon" achieve high classification accuracy, with only minor misclassifications, such as 3 instances of "Yellow Dragon" being misclassified as "Citrus Greening."

Some misclassifications are observed in other classes, though they are relatively minimal. For "Citrus Canker," 37 out of 43 instances are correctly classified, while 2 are misclassified as "Die Back," and 1 as "Spiny Whitefly," indicating some overlap in feature representation with these classes. Similarly, "Citrus Greening" misclassifies one instance as "Powdery Mildew," suggesting minor confusion between these categories.

The matrix also highlights areas for potential improvement. "Powdery Mildew" has 54 correct predictions out of 56, with one misclassified as "Citrus Greening." Additionally, the minority class "Yellow Dragon," which has only 13 instances, correctly predicts 10 but misclassifies 3 into "Citrus Greening." This suggests that further optimization or additional training data for minority classes could improve their classification performance.

In summary, the confusion matrix reflects the model's high effectiveness in classifying most disease and healthy leaf categories, with minimal errors and strong diagonal dominance. The few observed misclassifications are primarily associated with overlapping features or underrepresented classes, offering opportunities for refinement in these areas.

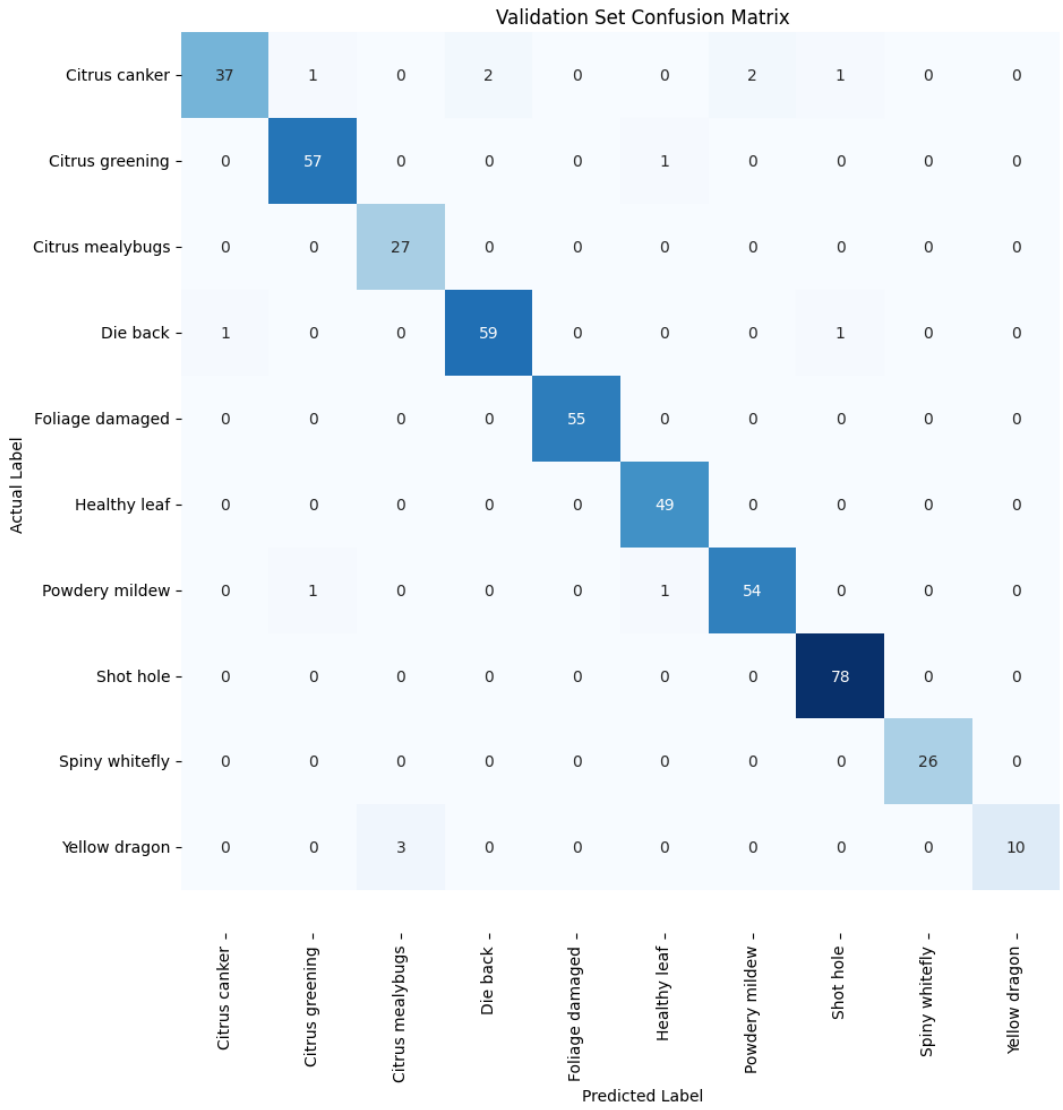

Figure 4.5: Confusion matrix of the validation set

The table presents the classification performance of the model on the test dataset, evaluated through precision, recall, F1-score, and support for each class. These metrics provide a detailed analysis of the model's ability to generalize to unseen data, highlighting its strengths and areas for improvement.

The model achieves excellent performance across most classes. Classes such as "Foliage Damaged," "Powdery Mildew," "Shot Hole," and "Spiny Whitefly" achieve perfect scores of 1.00 for precision, recall, and F1-score, indicating flawless classification with no misclassifications. Similarly, "Citrus Greening" and "Citrus Mealybugs" also show strong performance with F1-scores of 0.99 and 0.97, respectively, demonstrating the model's robustness in handling these categories.

Other classes, such as "Citrus Canker" and "Die Back," exhibit high performance with F1-scores of 0.96 and 0.95, respectively, although minor misclassifications are suggested by slightly lower precision or recall. "Healthy Leaf" achieves an F1-score of 0.97 with perfect

recall, reflecting the model's strong ability to correctly identify all instances of this class. The class "Yellow Dragon," which has the smallest support (5 instances), shows relatively lower precision (0.83) but perfect recall, leading to an F1-score of 0.91. This indicates that while all instances of "Yellow Dragon" are correctly identified, a few predictions might be false positives, likely due to its underrepresentation in the dataset.

Overall, the model achieves an accuracy of 98%, indicating that the vast majority of predictions are correct. The macro average F1-score is 0.97, reflecting consistent performance across all classes, while the weighted average F1-score of 0.98 demonstrates that the model performs well even when accounting for class imbalance. These results confirm the model's robustness and reliability in accurately classifying most disease and healthy leaf categories, with minimal errors and strong generalization to the test dataset.

Table 4.5: Test Set classification report

	Precision	Recall	F1 - Score	Support
Citrus Canker	0.94	0.97	0.96	34
Citrus Greening	1.00	0.97	0.99	35
Citrus Mealybugs	1.00	0.94	0.97	18
Die Back	0.97	0.93	0.95	40
Foliage Damaged	1.00	1.00	1.00	34
Healthy Leaf	0.94	1.00	0.97	34
Powdery Mildew	1.00	1.00	1.00	31
Shot Hole	1.00	1.00	1.00	42
Spiny Whitefly	1.00	1.00	1.00	15
Yellow Dragon	0.83	1.00	0.91	5
Accuracy			0.98	288
Macro Avg	0.97	0.98	0.97	288
Weighted Avg	0.98	0.98	0.98	288

The confusion matrix provides a comprehensive breakdown of the model's predictions on the test dataset by comparing actual labels to predicted labels for each class. The diagonal entries represent correct predictions, while the off-diagonal entries indicate misclassifications. Overall, the matrix demonstrates excellent performance, with most predictions falling along the diagonal, reflecting accurate classification.

Several classes achieve perfect classification, with all their instances correctly predicted. For example, "Foliage Damaged," "Healthy Leaf," "Powdery Mildew," "Shot Hole," and "Spiny Whitefly" have no misclassifications, showing the model's high precision and recall for these categories. These results highlight the robustness of the model in identifying these classes without errors.

Other classes, such as "Citrus Greening," "Citrus Canker," and "Die Back," show near-perfect performance with minimal misclassifications. For instance, 33 out of 34 instances of "Citrus Canker" are correctly classified, with one misclassified as "Die Back." Similarly, "Citrus Greening" has 34 out of 35 instances correctly identified, with one misclassified as "Citrus Canker." These misclassifications suggest slight overlaps in feature representations between these classes. Additionally, "Die Back" achieves 37 correct predictions out of 40, with two instances misclassified as "Foliage Damaged."

The minority class "Yellow Dragon," which has only 5 instances, is correctly classified for all instances, demonstrating the model's capability to handle underrepresented classes effectively in this test scenario.

Overall, the confusion matrix reflects the model's high accuracy and generalization on the test set. The minimal off-diagonal entries indicate that errors are rare and limited to a few specific categories with feature overlaps. These results confirm the model's strong reliability and effectiveness in classifying the test data with high precision and recall across all classes.

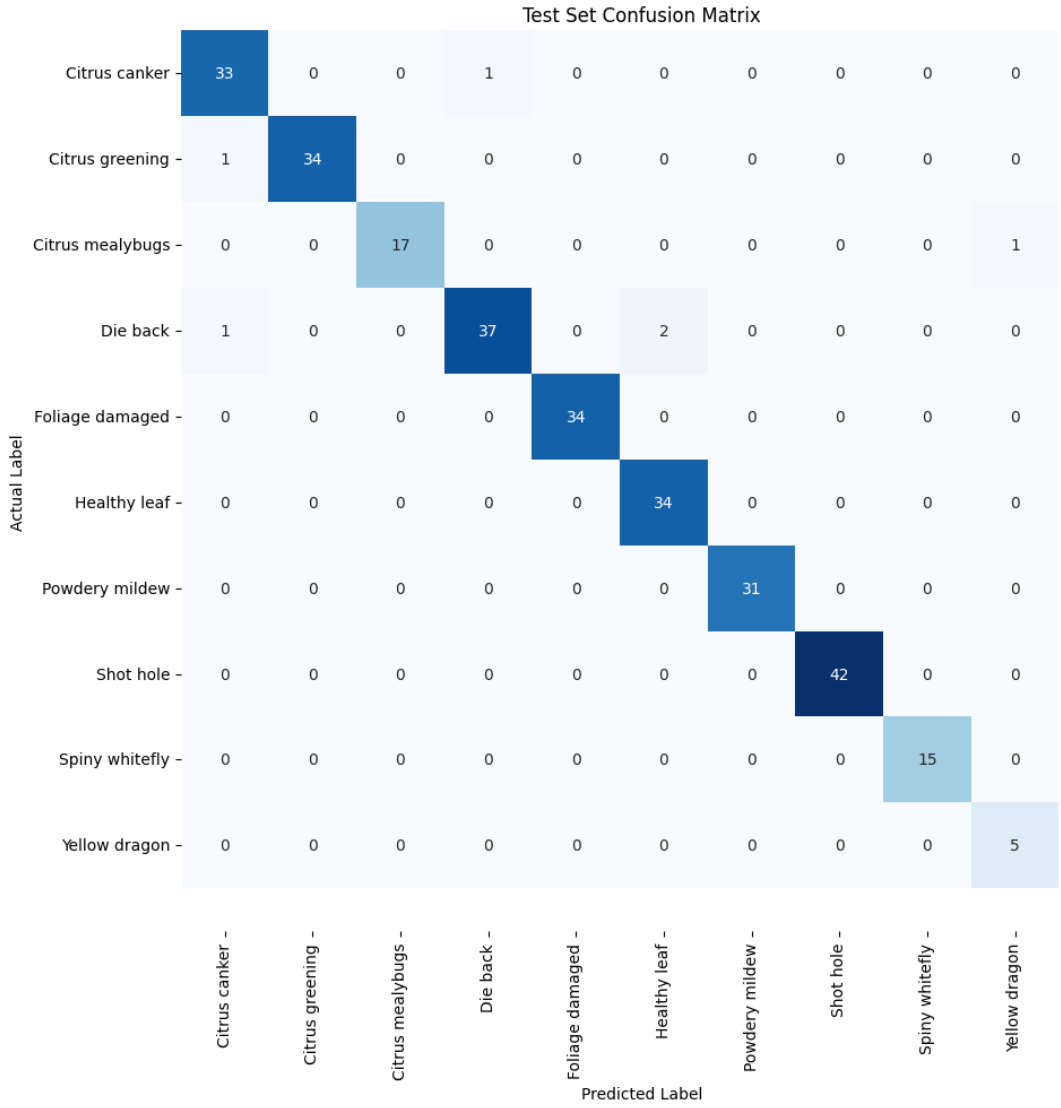

Figure 4.6: Confusion matrix of the test set

4.3.3 Performance Analysis of Hybrid Vision Transformer and Convolutional Neural Network (ViT-CNN) Model

The figure demonstrates the performance analysis of the hybrid Vision Transformer and Convolutional Neural Network (ViT-CNN) model using two metrics: accuracy and loss. The left plot displays the training and validation accuracy over 35 epochs, while the right plot illustrates the training and validation loss for the same period. These metrics highlight the model's learning dynamics and generalization capability.

In the accuracy plot, the training accuracy rapidly increases and stabilizes close to 100%, indicating the model's strong ability to learn from the training data. The validation accuracy also improves significantly during the initial epochs, stabilizing above 90%. However, the validation accuracy exhibits some fluctuations in later epochs, indicating

potential overfitting or sensitivity to the validation data. Despite these variations, the model achieves a consistently high validation accuracy.

The loss plot complements this analysis by showing the training and validation loss trends. The training loss decreases sharply in the initial epochs and stabilizes at a low value, demonstrating effective convergence during training. The validation loss decreases similarly but fluctuates in later epochs, aligning with the observed variations in validation accuracy. These fluctuations may suggest overfitting, as the model becomes more tuned to the training data while exhibiting minor instability on unseen validation data.

Overall, the ViT-CNN model achieves high performance with strong training and validation metrics. The occasional fluctuations in validation accuracy and loss suggest potential areas for further optimization, such as early stopping or regularization, to enhance stability and generalization. These results underscore the effectiveness of the hybrid ViT-CNN approach in leveraging the strengths of both architectures for the given dataset.

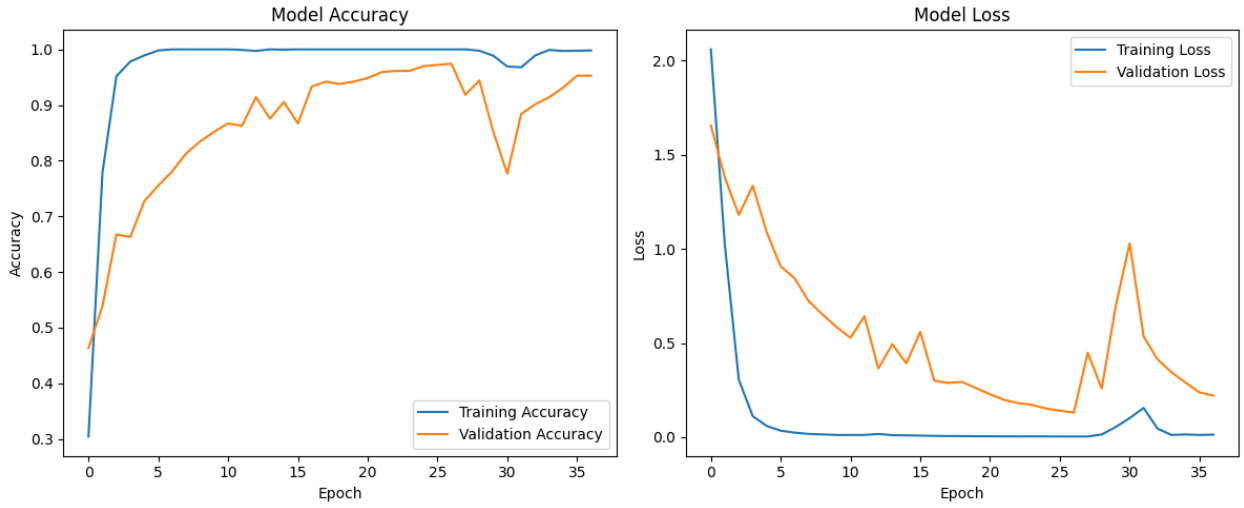

Figure 4.7: Validation Loss and accuracy curve of validation ViT-CNN model

The table presents the classification performance of the model on the validation dataset, evaluated through precision, recall, F1-score, and support for each class. The metrics indicate strong overall performance with high precision, recall, and F1-scores across most categories.

Several classes, such as "Citrus Canker," "Citrus Greening," "Die Back," "Shot Hole," "Spiny Whitefly," and "Yellow Dragon," achieve perfect precision (1.00), signifying no false positive predictions for these categories. The recall is also high for most of these classes, with values ranging from 0.91 to 1.00, resulting in strong F1-scores. For example, "Spiny Whitefly" achieves perfect precision, recall, and F1-score (1.00), indicating flawless

classification for this class.

The "Foliage Damaged" and "Powdery Mildew" classes exhibit strong recall (1.00), indicating that the model successfully identifies all instances of these categories. Their F1-scores are also high at 0.99 and 0.97, respectively, reflecting balanced precision and recall. Similarly, "Healthy Leaf" achieves a recall of 1.00 and an F1-score of 0.95, highlighting the model's reliability in detecting this class, although the precision is slightly lower at 0.91. "Yellow Dragon," despite having fewer instances (13), achieves high precision (1.00) and an F1-score of 0.96, although its recall is slightly lower at 0.92. This suggests that while the model avoids false positives for this minority class, a small number of actual instances were missed.

Overall, the model achieves an accuracy of 97%, indicating robust performance on the validation set. The macro average F1-score is 0.97, reflecting consistent performance across all classes, while the weighted average F1-score is also 0.97, accounting for the varying number of instances in each category. These results confirm the model's strong capability to classify diverse disease and healthy leaf categories with high precision and recall.

Table 4.6: Validation Set Classification Report

	Precision	Recall	F1 - Score	Support
Citrus Canker	1.00	0.91	0.95	43
Citrus Greening	1.00	0.97	0.96	58
Citrus Mealybugs	0.96	0.96	0.96	27
Die Back	1.00	0.93	0.97	61
Foliage Damaged	0.98	1.00	0.99	55
Healthy Leaf	0.91	1.00	0.95	49
Powdery Mildew	0.93	1.00	0.97	56
Shot Hole	0.99	1.00	0.99	78
Spiny Whitefly	1.00	1.00	1.00	26
Yellow Dragon	1.00	0.92	0.96	13
Accuracy			0.97	466
Macro Avg	0.98	0.97	0.97	466
Weighted Avg	0.98	0.97	0.97	466

The confusion matrix illustrates the model's performance on the validation dataset, displaying the actual versus predicted classifications for each class. The diagonal values represent correctly classified instances, while the off-diagonal values indicate misclassifications. The overall matrix demonstrates strong performance, with most

predictions concentrated along the diagonal, reflecting accurate classification.

Several classes achieve near-perfect classification. For instance, "Foliage Damaged," "Healthy Leaf," "Powdery Mildew," "Shot Hole," and "Spiny Whitefly" have no misclassifications, with all instances correctly predicted. This highlights the model's robust performance and precise identification for these categories. Similarly, "Citrus Mealybugs" performs well, with only one instance misclassified as "Powdery Mildew," resulting in strong accuracy for this class.

Other classes, such as "Citrus Canker" and "Citrus Greening," exhibit high accuracy but with minor misclassifications. For example, 39 out of 43 instances of "Citrus Canker" are correctly classified, with three instances misclassified as "Die Back" and one as "Powdery Mildew." Similarly, "Citrus Greening" correctly classifies 56 out of 58 instances, with one instance each misclassified as "Powdery Mildew" and "Citrus Mealybugs." These misclassifications could arise from overlapping features among these categories.

The "Yellow Dragon" class, which has the smallest support (13 instances), achieves 12 correct predictions, with one instance misclassified as "Citrus Mealybugs." Despite its limited representation, the model demonstrates strong performance for this minority class.

In summary, the confusion matrix highlights the model's strong generalization and classification capabilities across most classes. The few observed misclassifications are concentrated in categories with feature overlaps or fewer samples, suggesting potential areas for refinement. Overall, the matrix reaffirms the model's reliability and effectiveness in identifying diverse categories in the validation dataset.

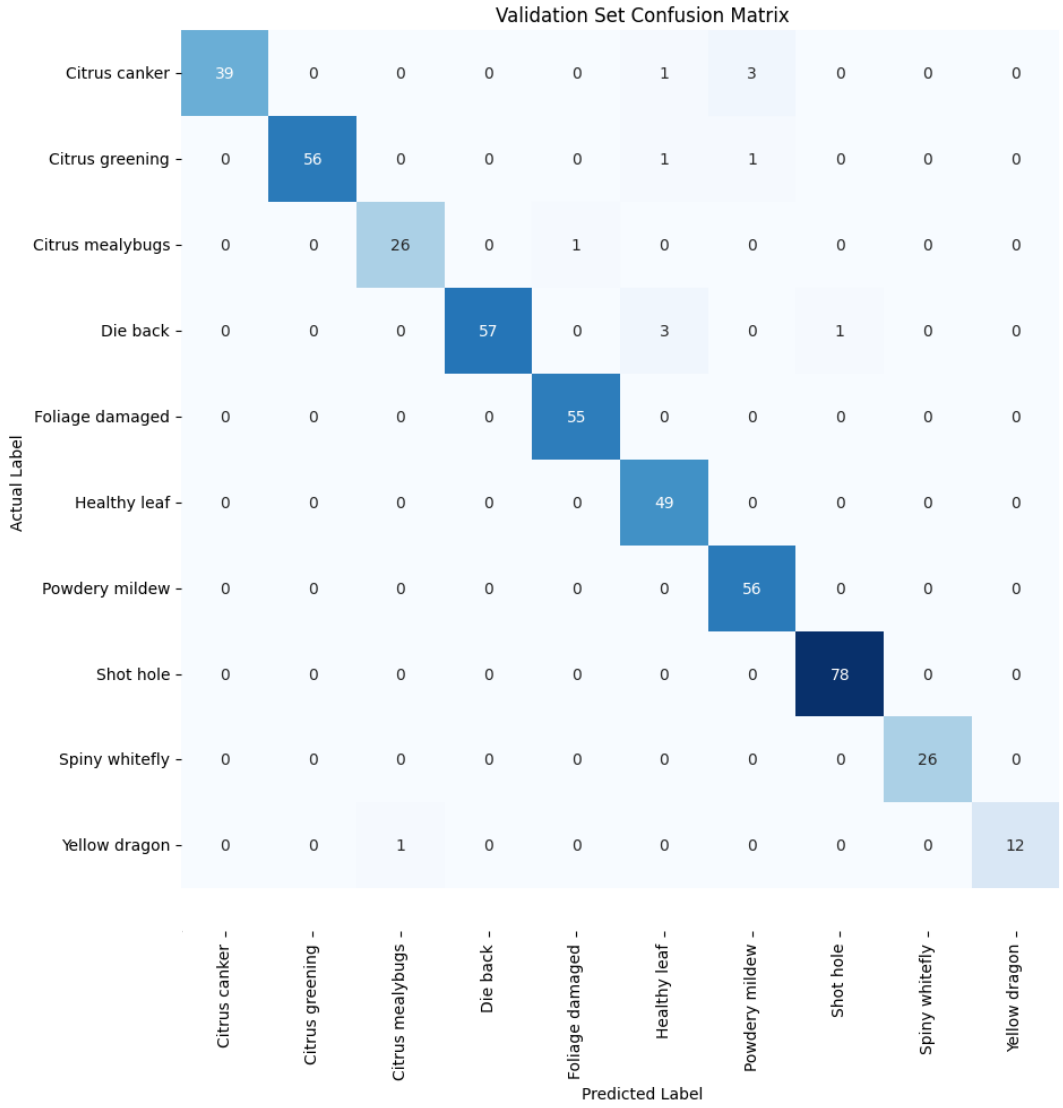

Figure 4.8: Confusion matrix of the validation set

The table presents the classification performance of the model on the test set, evaluated using precision, recall, F1-score, and support for each class. The results demonstrate high performance across all classes, with excellent precision, recall, and F1-scores, highlighting the model's robustness and generalization ability.

Several classes achieve perfect scores of 1.00 for precision, recall, and F1-score, including "Citrus Canker," "Powdery Mildew," "Shot Hole," "Spiny Whitefly," and "Yellow Dragon." These scores indicate that the model flawlessly identifies all instances of these categories with no false positives or false negatives. Similarly, "Foliage Damaged" and "Citrus Greening" exhibit high performance, with F1-scores of 0.99, reflecting strong overall accuracy for these classes.

Other classes, such as "Healthy Leaf" and "Die Back," perform well but show slightly lower precision or recall. For instance, "Healthy Leaf" achieves a recall of 1.00 but a precision of 0.92, resulting in an F1-score of 0.96, which suggests a few false positives. Similarly, "Die

Back" achieves a recall of 0.93 and an F1-score of 0.95, indicating that a small number of true instances were missed.

Overall, the model achieves an accuracy of 98%, with a macro average F1-score of 0.98 and a weighted average F1-score of 0.98. These metrics confirm the model's ability to maintain consistent performance across all classes, regardless of class imbalance or sample size. The high precision and recall across the majority of categories reflect the model's reliability and effectiveness in classifying diverse disease and healthy leaf categories in the test dataset.

Table 4.7: Test Set Classification Report

	Precision	Recall	F1 - Score	Support
Citrus Canker	1.00	1.00	1.00	34
Citrus Greening	0.97	1.00	0.99	35
Citrus Mealybugs	1.00	0.94	0.97	18
Die Back	0.97	0.93	0.95	40
Foliage Damaged	1.00	0.97	0.99	34
Healthy Leaf	0.92	1.00	0.96	34
Powdery Mildew	1.00	1.00	1.00	31
Shot Hole	1.00	1.00	1.00	42
Spiny Whitefly	1.00	1.00	1.00	15
Yellow Dragon	1.00	1.00	1.00	5
Accuracy			0.98	288
Macro Avg	0.99	0.98	0.98	288
Weighted Avg	0.98	0.98	0.98	288

The confusion matrix provides a detailed breakdown of the model's performance on the test set by comparing the actual labels with predicted labels. Each row represents the true class, and each column represents the predicted class. The diagonal values represent correctly classified instances, while the off-diagonal values indicate misclassifications. Overall, the matrix demonstrates excellent performance, with the majority of predictions correctly falling on the diagonal.

Several classes are classified perfectly, with all instances correctly predicted. These include "Citrus Canker," "Citrus Greening," "Healthy Leaf," "Powdery Mildew," "Shot Hole," "Spiny Whitefly," and "Yellow Dragon." These results highlight the model's exceptional ability to identify these categories without any misclassifications.

Minor misclassifications are observed in a few classes. For example, "Citrus Mealybugs" has 17 correct predictions out of 18 instances, with one instance misclassified as "Die

Back." Similarly, "Die Back" shows three misclassifications, where 37 instances are correctly predicted, and three are incorrectly classified as "Foliage Damaged." Additionally, "Foliage Damaged" has one misclassification, with 33 out of 34 instances correctly predicted, and one instance classified as "Citrus Greening." These errors could be attributed to overlapping features or similarities in the visual characteristics of these categories.

Overall, the confusion matrix underscores the model's strong classification performance, with only a small number of errors in a few specific categories. These results confirm the robustness of the model in handling a diverse set of classes, with minimal misclassifications that could be further reduced with additional fine-tuning or targeted feature extraction for challenging categories.

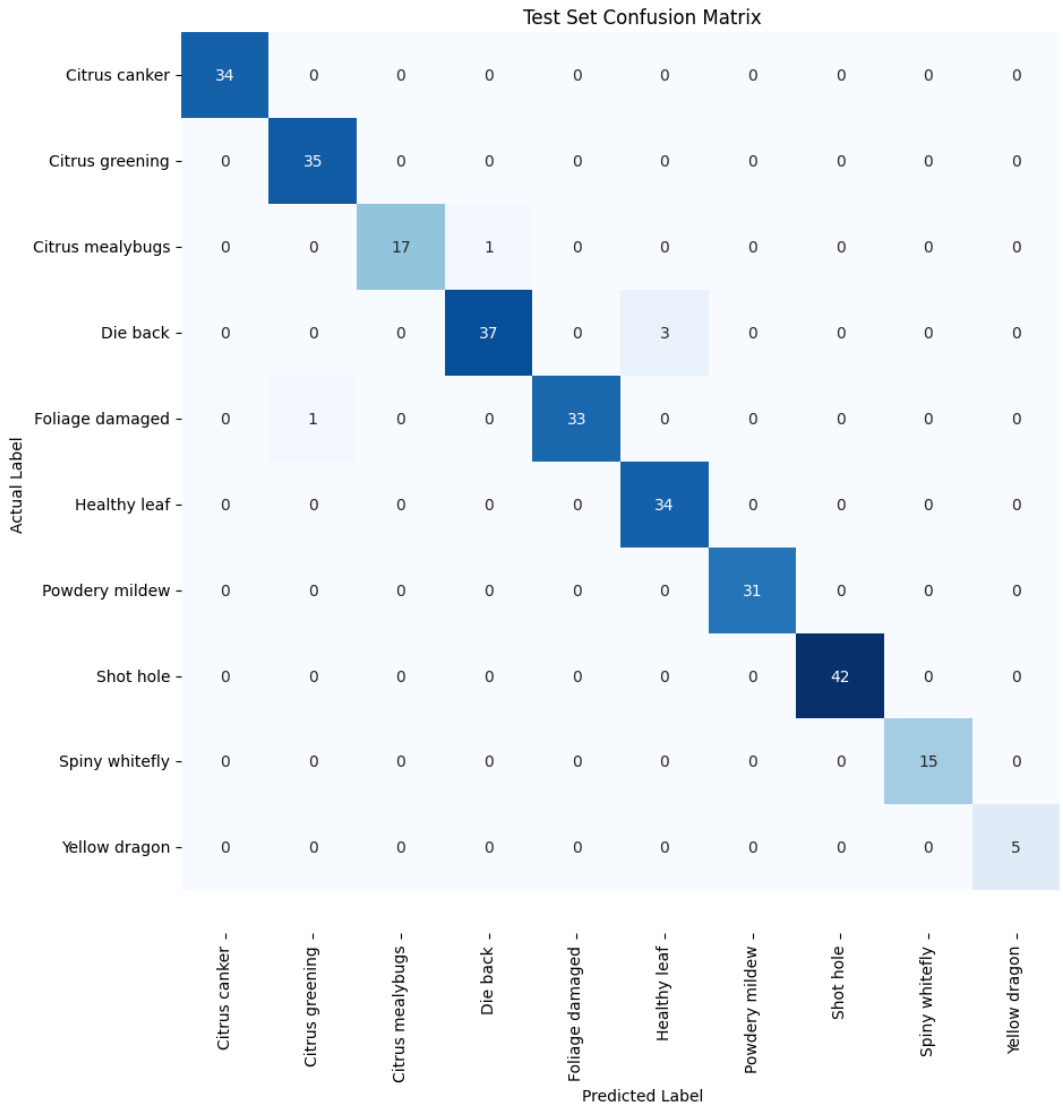

Figure 4.9: Confusion matrix of the test set

4.4 Summary

This chapter analyzed the performance of three experimented models—Vision Transformer (ViT), ResNet50v2, and the hybrid ViT-CNN—on the classification task. The evaluation included metrics such as accuracy, precision, recall, F1-score, and confusion matrices to assess the models' learning and generalization capabilities.

The ViT model showed moderate performance with training, validation, and test accuracies of 92.78%, 90%, and 89.9%, respectively, indicating reasonable learning but limited generalization. The ResNet50v2 model demonstrated consistent and robust performance, achieving accuracies of 97.92% across training and test datasets, and 97% on validation data. Finally, the ViT-CNN hybrid model outperformed the others, achieving the highest test accuracy of 98.26%, along with strong precision, recall, and F1-scores across all classes, making it the most effective model for the task.

The analyses revealed that the ViT-CNN hybrid model excelled in leveraging the strengths of both Vision Transformers and CNNs, offering superior generalization and minimal misclassifications. While ResNet50v2 also performed reliably, the ViT model showed room for improvement. The chapter concludes that the ViT-CNN hybrid model is the optimal choice for this classification task.

Chapter 5

Engineering Standards and Design Challenges

This chapter discusses the engineering standards adhered to during the project and the challenges encountered during the design process. It emphasizes the software, hardware, and communication standards, followed by an exploration of the societal, environmental, and ethical impacts. Finally, it provides a sustainability plan and presents the complexity of the problem and how it was solved.

5.1 Compliance with the Standards

Late diagnosis and rapid disease progression often result in poor survival rates. OSCC accounts for the majority of oral cancer cases, and histopathological examination remains the most reliable method for accurate diagnosis. However, traditional diagnostic approaches are labour-intensive, subject to inter-observer variability, and require expert pathologists, making them impractical for large-scale screening. These challenges underscore the urgent need for automated, reliable, and accurate diagnostic tools for early detection and intervention. Recent advancements in artificial intelligence (AI) and deep learning have shown immense potential in revolutionizing medical imaging analysis, particularly for cancer diagnosis. Convolutional Neural Networks (CNNs) have emerged as the backbone of modern computer-aided diagnosis (CAD) systems, offering automated feature extraction and high accuracy.

5.1.1 Software Standards

The implementation of this thesis adheres to widely recognized software standards, ensuring the use of reliable, efficient, and scalable tools for the development of the proposed system. The following software components were utilized:

Platform: Google Colaboratory, a cloud-based Jupyter Notebook environment, was used for all experimentation and model development, ensuring seamless integration with the cloud infrastructure.

Programming Language: Python 3.x was chosen for its robust ecosystem of libraries and tools suited for deep learning and machine learning tasks.

Libraries:

- TensorFlow: Used for the development of CNN and Vision Transformer (ViT) models.
- Keras: Integrated within TensorFlow to simplify model building and training.
- TorchVision: Employed for specific experiments requiring PyTorch, providing tools for image transformations and model development.
- NumPy and Pandas: Utilized for efficient data handling and preprocessing tasks.
- OpenCV: Used for advanced image preprocessing and augmentation techniques.
- Matplotlib and Seaborn: For generating insightful visualizations of data and model performance.

Dataset Storage: Google Drive was integrated into Colab to enable efficient dataset storage and accessibility across experiments.

5.1.2 Hardware Standards

The research was conducted on a cloud-based hardware infrastructure provided by Google Colaboratory, which offers the following standardized hardware specifications:

- Processor: Up to 2 virtual CPU cores in the free tier; additional processing power is available in Colab Pro.
- GPU: Access to NVIDIA GPUs, including Tesla K80, T4, P100, or V100, depending on resource allocation and subscription tier.
- RAM: 12 GB of RAM for free-tier users, with up to 25 GB available for Colab Pro subscribers.

- Disk Storage: Temporary runtime storage of up to 15 GB was utilized during experimentation for loading datasets and saving intermediate results.

The use of a cloud-based infrastructure ensured scalability, cost-efficiency, and the ability to handle the computational requirements of deep learning model training and evaluation.

5.1.3 Communication Standards

Communication and data transfer within this research adhered to secure and reliable standards provided by Google's cloud infrastructure. The key aspects include:

- Google Drive Integration: Data storage and retrieval were facilitated through Google Drive, ensuring secure, encrypted access to datasets and model checkpoints during the experiments.
- Runtime Connectivity: Google Colab's runtime maintained a secure HTTPS connection, ensuring the safety and privacy of data during processing and communication between the local system and the cloud environment.
- Collaborative Features: Google Colab's real-time collaboration functionality was utilized for sharing code, results, and insights with supervisors or peers, enhancing team communication and ensuring compliance with project timelines.

These communication standards provided a secure, reliable, and collaborative framework for conducting and sharing research outcomes.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The implementation of an AI-based system for detecting and classifying sweet orange leaf diseases has the potential to significantly improve the quality of life for farmers and agricultural workers. By providing timely and accurate identification of diseases, this system enables farmers to make informed decisions about disease management, reducing crop losses and improving yield quality. Early detection ensures that appropriate interventions, such as targeted pesticide use or crop rotation, can be

implemented, minimizing the spread of diseases and associated economic losses.

For small-scale farmers, particularly in resource-constrained regions, this technology can be a game-changer. By integrating lightweight and efficient models, the system can be deployed on affordable devices, such as smartphones or IoT-enabled cameras, making advanced disease detection accessible to a broader population. This accessibility enhances the livelihoods of farmers by increasing productivity and reducing labor-intensive manual inspections. Additionally, by minimizing reliance on guesswork and outdated practices, this system helps reduce stress and uncertainty, promoting mental well-being among farmers.

Furthermore, the improved crop quality resulting from precise disease detection contributes to the health and nutrition of consumers. High-quality sweet oranges free from disease ensure better taste, longer shelf life, and higher nutritional value, benefiting the broader population.

5.2.2 Impact on Society & Environment

The societal impact of this research lies in its potential to support sustainable agricultural practices and ensure food security. By reducing crop losses and optimizing disease management, the system can help stabilize the supply chain for sweet oranges, which are a vital citrus crop globally. This stability can positively influence market prices and economic growth in agricultural communities, particularly in regions heavily reliant on citrus exports, such as Bangladesh.

On an environmental level, the system promotes eco-friendly farming practices. Accurate disease detection enables targeted pesticide applications, significantly reducing overuse and its harmful effects on soil, water, and non-target organisms. This contributes to preserving biodiversity and maintaining ecological balance in agricultural areas. Additionally, by improving crop health and reducing wastage, the system indirectly lowers the carbon footprint associated with replanting, overproduction, and transportation of substandard crops.

Moreover, the scalability of this technology encourages its adoption for other crops, fostering widespread sustainability in agriculture. By addressing critical challenges in crop management, the research aligns with global goals for sustainable development, including zero hunger, responsible consumption and production, and climate action.

In summary, the proposed system positively influences individual lives, supports

societal growth, and fosters environmental sustainability. It equips farmers with advanced tools to combat diseases while contributing to the broader goals of sustainable agriculture and environmental protection.

5.2.3 Ethical Aspects

The deployment of AI-based tools for sweet orange leaf disease detection raises important ethical considerations. Foremost among these is ensuring equitable access to the technology. Farmers, especially those in resource-constrained regions, may face barriers to adopting advanced AI systems due to limited technical skills or financial constraints. To address this, the research emphasizes the development of lightweight and affordable models that can be deployed on commonly available devices such as smartphones.

Another ethical concern lies in data collection and usage. Protecting the privacy and ownership of data collected from agricultural fields is crucial. Farmers must be informed about how their data is used, and consent mechanisms should be incorporated into any deployment strategy. Additionally, transparency in AI decision-making is vital. By integrating Explainable AI (XAI) techniques, this research ensures that the models provide interpretable results, empowering farmers to trust and understand the recommendations provided.x$

Finally, the potential misuse of such technology, such as reliance on AI outputs without human oversight, should be mitigated. The system is designed to act as a decision-support tool, augmenting but not replacing the expertise of agricultural professionals. This ensures that ethical standards in technology deployment are upheld while maintaining human accountability.

5.2.4 Sustainability Plan

The sustainability of this research lies in its focus on long-term usability and scalability. The following strategies outline the sustainability plan:

1. The models developed in this study are optimized for efficiency and scalability. By using lightweight architectures, such as hybrid ViT-CNN models, the system can operate on low-power devices, ensuring long-term technological relevance and adaptability to advancements in hardware and software.

2. The cost-effectiveness of the system makes it accessible to small-scale farmers, ensuring widespread adoption. Additionally, reducing crop losses and improving yields through early disease detection generates economic benefits that can sustain the deployment and maintenance of the technology.

3. By enabling precise and reduced pesticide usage, the system minimizes environmental harm, contributing to the preservation of soil health, water quality, and biodiversity. The system's alignment with eco-friendly farming practices ensures its contribution to sustainable agricultural ecosystems.

4. The sustainability plan includes training programs for farmers and agricultural workers to ensure they can effectively use the system. Community-based workshops and collaborations with agricultural organizations can facilitate knowledge transfer and long-term adoption.

5. The system is designed to be scalable and adaptable to other crops and regions. Its modular architecture allows for easy customization, ensuring relevance beyond sweet orange crops and fostering sustainability in broader agricultural contexts.

5.3 Project Management and Financial Analysis

Provide a cost analysis in terms of the required budget and revenue model. In the case of budget, you must show an alternate budget and rationales.

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

Table 5.1 provides a detailed mapping of the research problem to the problem-solving categories. It demonstrates how the project addresses key aspects such as depth of knowledge, conflicting requirements, and stakeholder involvement.

Table 5.1: Mapping with complex problem-solving.

EP1 Dept of Knowledge	EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stake-holder Involvement	EP7 Interdependence
Deep understanding of CNN, ViT, and hybrid models for disease detection	Balancing accuracy, computational efficiency, and dataset quality	Evaluating models using accuracy, F1-score, and recall metrics	Addressing dataset limitations and scalability issues	Following best practices in TensorFlow and PyTorch usage	Farmers and agricultural experts as primary beneficiaries	Integration of preprocessing, training, and evaluation workflows

Mapping with Knowledge Profile for EP1

Table 5.2 maps the Depth of Knowledge (EP1) to the Knowledge Profile categories. It illustrates the application of engineering fundamentals, advanced techniques, and research literature in the project.

Table 5.2: Mapping with knowledge Profile.

K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K8 Research Literature
Application of machine learning and computer vision principles	Advanced techniques like hybrid CNN-ViT models	Workflow design from data preprocessing to evaluation	Implementation using cloud-based Google Colab platform	Building the foundation through an extensive literature review

5.4.2 Engineering Activities

This section provides a mapping with engineering activities. Each mapping highlights the activities undertaken as part of the research and provides a rationale for their inclusion.

Table 5.3 highlights the complex engineering activities involved in the research, such as utilizing cloud resources, fostering collaboration, introducing innovative hybrid models, and addressing societal and environmental impacts. It emphasizes the familiarity with cutting-edge frameworks.

Table 5.3: Mapping with complex engineering activities.

EA1 Range of Resources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for Society and Environment	EA5 Familiarity
Utilization of Google Colab's cloud-based GPU resources for efficient model training.	Collaboration with agricultural experts for real-world validation.	Integration of hybrid CNN-ViT models for innovative solutions.	Reduction in pesticide overuse and environmental harm.	Familiarity with TensorFlow and PyTorch frameworks.

5.5 Summary

This chapter comprehensively addresses the engineering aspects of the project. It explains the implementation of software, hardware, and communication standards to ensure efficient and scalable deployment. The societal and environmental impacts are analyzed, highlighting the benefits to farmers and the ecosystem. Ethical considerations are discussed to promote fair use and inclusivity of the system. A sustainability plan ensures the long-term adaptability and usability of the project. Finally, the complex engineering problem is mapped and rationalized, outlining the innovative approaches adopted in this research.

Chapter 6

Conclusion

This chapter concludes the research by summarizing the key findings, addressing the limitations of the study, and proposing directions for future work. It highlights the contributions made toward improving sweet orange disease detection using advanced deep learning techniques while identifying areas for further exploration and development.

6.1 Summary

This study investigated deep learning models, including Vision Transformers (ViT), Convolutional Neural Networks (CNN), and hybrid ViT-CNN architectures, for detecting and classifying sweet orange leaf diseases. The research demonstrated the potential of these models to achieve high accuracy in identifying diseases, significantly improving upon traditional manual methods. (Include accuracy metrics here.)

The integration of Explainable Artificial Intelligence (XAI) techniques, such as Grad-CAM, enhanced the interpretability of the models, making them more user-friendly for farmers and agricultural experts. Additionally, the study addressed challenges related to dataset scarcity by applying preprocessing techniques like contrast boosting and segmentation, ensuring the models were trained on robust and diverse data. The development of lightweight and efficient hybrid models ensured that the solutions were deployable in resource-constrained environments, such as small-scale farms.

The findings contribute to sustainable agriculture by enabling precise disease detection, minimizing crop losses, and reducing the environmental impact of excessive pesticide use. By addressing key gaps in existing research, this study provides a strong foundation for further advancements in AI-driven agricultural solutions.

6.2 Limitation

While this research achieved significant advancements, it is important to acknowledge the following limitations:

1. The study relied on a limited dataset for training and testing, which may not

capture the full diversity of sweet orange leaf diseases across different regions and environmental conditions.

2. Although the models performed well in controlled experiments, they were not extensively tested under field conditions, which may present additional challenges such as variable lighting, occlusion, or noise.
3. Advanced models like ViT and hybrid architectures require substantial computational resources, potentially limiting their accessibility for resource-constrained farmers. While lightweight models were developed, further optimisation is needed for wider adoption.
4. This study does not present any conflicts of interest. However, it is essential to note that future collaborations with commercial agricultural organizations or technology providers must ensure unbiased and transparent use of the developed models.

6.3 Future Work

Building on the findings of this study, the following areas are suggested for future research:

1. While this study focused on sweet orange leaf diseases, future research could adapt and test the developed models on other citrus crops and a broader range of agricultural products to enhance the generalizability of the system.
2. Future work could focus on deploying the system in real-world agricultural environments and testing its performance under varying conditions, such as different lighting, occlusion, and disease progression stages.
3. Incorporating the models into IoT-enabled devices, such as drones and smart cameras, could facilitate automated and large-scale disease detection in agricultural fields.
4. Collaborating with agricultural institutions to create larger and more diverse datasets specific to sweet orange diseases would further enhance model performance and reliability.
5. Developing more sophisticated XAI methods to improve the interpretability of model decisions would strengthen trust and usability among farmers and agricultural professionals.
6. Further optimization to reduce computational requirements and latency could enable seamless deployment on edge devices, such as smartphones and low-power hardware.

References

[1] Khattak, A., Habib, A., Asghar, M. Z., Subhan, F., Razzak, I., & Habib, A. (2021). Applying deep neural networks for user intention identification. Soft Computing, 25, 2191-2220.

[2] Lanjewar, M. G., & Parab, J. S. (2023). CNN and transfer learning methods with augmentation for citrus leaf diseases detection using PaaS cloud on mobile. Multimedia Tools and Applications, 1-26.

[3] Boukabouya, R. A., Moussaoui, A., & Berrimi, M. (2022, November). Vision Transformer Based Models for Plant Disease Detection and Diagnosis. In 2022 5th International Symposium on Informatics and its Applications (ISIA) (pp. 1-6). IEEE.

[4] Dümen, S., Yılmaz, E. K., Adem, K., & Avaroglu, E. (2023). Achieving High Accuracy in Lemon Quality Classification: A Comparative Study of Deep Learning and Transformer Models.

[5] Thakur, P. S., Khanna, P., Sheorey, T., & Ojha, A. (2021, December). Vision transformer for plant disease detection: PlantViT. In International Conference on Computer Vision and Image Processing (pp. 501-511). Cham: Springer International Publishing.

[6] Yong, W. C., Ng, K. W., Haw, S. C., Naveen, P., & Ng, S. B. (2024). Leaf Condition Analysis Using Convolutional Neural Network and Vision Transformer. International Journal of Computing and Digital Systems, 16(1), 1-10.

[7] da Silva, J. C., Silva, M. C., Luz, E. J., Delabrida, S., & Oliveira, R. A. (2023). Using Mobile Edge AI to Detect and Map Diseases in Citrus Orchards. Sensors, 23(4), 2165.

[8] Prashanthi, B., Krishna, A. V., & Rao, C. M. (2024). LEViT-Leaf Disease identification and classification using an enhanced Vision transformers (ViT) model. Multimedia Tools and Applications, 1-32.

[9] Gupta, M., Sharma, S. K., & Sampada, G. C. (2023). Classification of Brain Tumor Images Using CNN. Computational Intelligence and Neuroscience, 2023.

[10] Emon, S. H., Islam, I. K., Nahin, T. J., & Ahmed, A. M. (2023). An efficient deep learning approach to detect citrus leaves disease (Doctoral dissertation, Brac University).

[11] De Silva, M., & Brown, D. (2023). Multispectral Plant Disease Detection with Vision Transformer–Convolutional Neural Network Hybrid Approaches. Sensors, 23(20), 8531.

[12] Garg, S., & Krishnamurthi, R. (2023). A CNN encoder decoder LSTM model for sustainable wind power predictive analytics. Sustainable Computing: Informatics and Systems, 38, 100869.

[13] Momeny, M., Neshat, A. A., Jahanbakhshi, A., Mahmoudi, M., Ampatzidis, Y., & Radeva, P. (2023). Grading and fraud detection of saffron via learning-to-augment incorporated Inception-v4 CNN. Food Control, 147, 109554.

[14] Pourdarbani, R., Sabzi, S., Dehghankar, M., Rohban, M. H., & Arribas, J. I. (2023). Examination of Lemon Bruising Using Different CNN-Based Classifiers and Local Spectral-Spatial Hyperspectral Imaging. Algorithms, 16(2), 113.

[15] Thai, H. T., Le, K. H., & Nguyen, N. L. T. (2023). FormerLeaf: An efficient vision transformer for Cassava Leaf Disease detection. Computers and Electronics in Agriculture, 204, 107518.
[16] Lee, S., Choi, G., Park, H. C., & Choi, C. (2022). Automatic Classification Service System for Citrus Pest Recognition Based on Deep Learning. Sensors, 22(22), 8911.
[17] Wu, S., Sun, Y., & Huang, H. (2021, December). Multi-granularity feature extraction based on vision transformer for tomato leaf disease recognition. In 2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST) (pp. 387-390). IEEE.
[18] Sudha, N., & Vignesh, B. S. (2024). Quantum-enhanced Diagnosis: Revolutionizing Tomato Leaf Disease Detection. Nanotechnology Perceptions, 798-818.
[19] Fahim-Ul-Islam, M., Chakrabarty, A., Ahmed, S. T., Rahman, R., Kwon, H. H., & Piran, M. J. (2024). A Comprehensive Approach Towards Wheat Leaf Disease Identification Leveraging Transformer Models and Federated Learning. IEEE Access.
[20] Sanyal, S., Adhikary, R., & Choudhury, S. J. (2024). Revolutionizing lemon grading: an automated CNN-based approach for enhanced quality assessment. International Journal of Information Technology, 1-12.
[21] Pramanik, A., Khan, A. Z., Biswas, A. A., & Rahman, M. (2021, July). Lemon leaf disease classification using CNN-based architectures with transfer learning. In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

11/01/25

Hybrid Deep Learning Approach for Sweet Orange Leaf Disease Detection Using CNN and Vision Transformers

ORIGINALITY REPORT

25%	14%	16%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	3%
2	Submitted to United International University Student Paper	1%
3	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
4	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1%
5	"Proceedings of the 5th International Conference on Data Science, Machine Learning and Applications; Volume 1", Springer Science and Business Media LLC, 2025 Publication	1%
6	Submitted to Southern Arkansas University (Blackboard LTI 1.3) Student Paper	1%
7	arxiv.org Internet Source	