Classification of Okra Leaf Diseases Using Machine Learning

By
Md. Mahfuz Ahmed
ID: 211-15-4034

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. Shahadat Hossain
(Assistant Professor)

Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

Co-Supervisor

Mr. Md Jihan Parvaj
(Lecturer)

Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

January 12, 2025

APPROVAL

This Project titled "**Automated Detection and Classification of Okra Leaf Diseases Using Machine Learning**," submitted by [**Md. Mahfuz Ahmed**] and 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 **12-01-2025**.

BOARD OF EXAMINERS

Dr. Md. Zahid Hasan (ZH)
Board Chairman
Associate Professor, Chairman,
Department of CSE, FSIT
Daffodil International
University

Dr. Md Alamgir Kabir (DMAK)
Internal Examiner 1
Assistant Professor,
Department of CSE, FSIT
Daffodil International
University

Mr. Tanvirul Islam (TI)
Internal Examiner 2
Lecturer,
Department of CSE, FSIT
Daffodil International
University

Dr. Md. Manowarul Islam
External Examiner
Associate Professor,
Department of CSE
Jagannath University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mr. Shahadat Hossain (Assistant Professor)**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

11.01.25

Mr. Shahadat Hossain
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Mr. Md Jihan Parvaj
Lecturer
Department of Computer Science and Engineering
Daffodil International University

Submitted by:

Mahfuz Ahmed
12.01.25

Md. Mahfuz Ahmed
Student ID: 211-15-4034
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 as- sisted 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. Shahadat Hossain**, **(Assistant Professor)**, 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** to carry out this project. His endless pa- tience, 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 Com- puter 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

In precision agriculture, machine learning models are increasingly revolutionizing disease detection and management for crops. Recent advances in computer vision, particularly Vision Transformer (VIT) architectures, have shown promising results in accurately identifying plant diseases. This study explores three VIT model variants: Pre-trained VIT, Mobile VIT, and Scratch VIT applied to Okra leaf disease detection, an area where effective early identification can significantly improve crop yield. The dataset, containing 3,775 Okra leaf images includes four disease classes and represents diverse environmental conditions to ensure robust model performance. Among the models, the Pre-trained VIT demonstrated the highest performance, achieving 96% validation and test accuracy with an AUC score of 0.985, indicating strong generalizability and minimal overfitting. Scratch VIT followed closely with 93% validation accuracy, 94% test accuracy, and a 0.98 AUC score, showcasing reliable classification despite being trained from scratch. Mobile VIT achieved 83% validation and 85% test accuracy with an AUC of 0.962, suggesting some limitations in handling complex features. The study highlights the Pre-trained VIT's potential as a reliable and efficient solution for okra disease detection, providing an effective tool for farmers to make informed, proactive decisions in crop management. This approach emphasizes the value of VIT models in advancing precision agriculture through accurate, disease classification.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Agriculture remains a critical sector worldwide, supporting economic stability and food security. Okra (Abelmoschus esculentus), commonly cultivated in tropical and subtropical regions, plays a significant role in this sector, particularly in Asia and Africa. However, okra cultivation faces considerable challenges from various diseases, with serious implications for crop quality and yield. Among these diseases, the Yellow Vein Mosaic Virus (YVMV) is notably destructive, transmitted by whiteflies, and capable of reducing yields by up to 94% under severe infestation [1]. The increasing prevalence of other pathogens, such as Cercospora Leaf Spot and Powdery Mildew, further highlights the urgent need for precise and timely disease management strategies [2].

Traditional agriculture often relies on manual inspection for disease detection, a process that is time-consuming, prone to human error, and dependent on agricultural expertise. In many rural areas, farmers lack access to plant pathology resources, exacerbating the challenge [3]. In response, the agricultural sector has increasingly adopted and data-driven approaches for disease identification. Machine learning (ML) and deep learning (DL) techniques have emerged as promising solutions, enabling accurate and efficient disease detection through image analysis [4]. Convolutional Neural Networks (CNNs), for example, have demonstrated remarkable capabilities in recognizing plant diseases by analyzing visual patterns in leaf images [2], [5]. By minimizing human intervention, these technologies are revolutionizing traditional agricultural practices.

The current state of research showcases diverse approaches to okra leaf disease detection using ML. Hybrid models such as CNN-Random Forest [2] and CNN-KNN combinations [6] have achieved impressive accuracy, effectively addressing diseases like YVMV and Anthracnose. Federated learning models, as developed by Jindal [5], offer privacy-preserving classification by enabling decentralized data contribution, while transfer learning techniques have allowed researchers to leverage pre-trained networks for okra disease classification even with limited datasets. For instance, Hridoy [7] employed InceptionResNetV2 and Res Net architectures to manage a dataset of 124,760 images, illustrating the scalability of transfer learning. Similarly, Kumar [8] utilized YOLOv8 variants for high-speed, high-accuracy disease detection, demonstrating the potential for real-time monitoring.

Despite these advancements, significant challenges remain. Many models are developed and tested in controlled environments, which may not account for the environmental variability of real-world agricultural settings [9]. Furthermore, while high-accuracy models like ResNet50 have proven effective [10], their computational demands limit feasibility in resource-constrained environments [11]. Addressing these gaps is vital for the broader adoption of ML-based solutions in agriculture.

1.2 Motivation

Okra, a nutritionally and economically valuable crop, faces significant threats from various plant diseases, with Yellow Vein Mosaic Virus (YVMV) being the most severe. Capable of reducing yields by up to 94% in severe cases, YVMV disrupts the chlorophyll content and overall health of plants, presenting a persistent challenge for farmers [1]. Other diseases, such as Cercospora Leaf Spot and Powdery Mildew, exacerbate this challenge. Currently, farmers rely heavily on visual assessments and manual inspections, methods that are time-intensive and prone to inaccuracies due to human error [2-3].

Advancements in machine learning (ML) and deep learning (DL) have shown promise in addressing these challenges. CNNs and hybrid models have demonstrated considerable accuracy in detecting okra diseases [4], [6], but their real-world applicability remains limited by several factors. High-computation models like ResNet50, while accurate, are unsuitable for deployment in rural farming environments with limited technological resources [10]. Moreover, these models are often trained on controlled datasets, lacking the robustness to handle real-world environmental variability [9].

Privacy concerns also hinder the adoption of centralized data-driven solutions in data-sensitive regions. Federated learning, which enables decentralized model training, has been proposed to address these issues, but its application remains limited [5]. Additionally, existing studies often focus on a narrow range of diseases, primarily YVMV, leaving other diseases and multi-disease scenarios underexplored [2], [12].

This narrow scope limits the utility of these models in broader agricultural contexts.

To bridge these gaps, there is a pressing need for cost-effective, adaptable, and privacy-conscious solutions that can reliably operate in real-world agricultural settings. Comprehensive ML models capable of identifying multiple diseases across diverse environmental conditions are essential for empowering farmers, enhancing crop health, and ensuring sustainable yields in resource-constrained environments.

1.3 Objectives

The primary objective of this study is to develop a robust, adaptable, and efficient machine learning model based on Vision Transformer (VIT) architectures for the classification of okra leaf diseases. This transformer-based approach aims to enhance agricultural practices by equipping farmers with a reliable tool for early disease diagnosis, ultimately improving crop yield and quality. The specific objectives of this study are as follows:

Implement and evaluate various VIT architectures, including Mobile VIT, Pre-trained VIT, and VIT from scratch, to accurately detect multiple okra leaf diseases. This approach addresses the limitations of conventional models by leveraging the attention mechanism inherent in transformers for improved disease detection [2], [6].

Design the VIT models with adaptability to account for environmental variations in real-world agricultural settings, ensuring robustness and generalizability outside controlled datasets.

Customize and optimize the VIT models to function efficiently on low-resource hardware, making the technology accessible to farmers in rural and economically constrained areas where high-performance computational resources may not be readily available.
Ensure the VIT-based model can handle multi-disease scenarios effectively, broadening its applicability across various agricultural contexts for comprehensive okra disease management.

Design the VIT model's output to deliver clear, actionable information regarding disease severity and recommended interventions, empowering farmers to make informed decisions and proactively manage crop health.

1.4 Methodology

The proposed methodology for okra leaf disease detection combines data preprocessing, advanced deep learning models, and user-friendly application development. The following steps outline the process:

Dataset Preparation: An extensive dataset of okra leaf images is utilized, encompassing various diseases such as Yellow Vein Mosaic Virus (YVMV), Cercospora Leaf Spot, and Powdery Mildew.

Preprocessing: The raw images undergo preprocessing steps, including normalization, image resizing, and adjustment of dots per inch (DPI), to ensure uniformity and compatibility with deep learning models.

Model Selection and Training: Vision Transformer (VIT) models form the backbone of the disease detection system. The study evaluates three configurations: Scratch VIT, Pre-Trained VIT, and Mobile VIT. These models are compared to determine the most efficient and accurate architecture for disease classification.

Model Optimization: The best-performing VIT model is converted into a Tensor Flow Lite (TF Lite) format for deployment, ensuring compatibility with resource-constrained environments.

Application Development: A mobile application is developed using Flutter to serve as an interface for end-users. The app enables farmers to upload leaf images, detect diseases, and receive detailed reports on the health of their crops.

Evaluation and Comparison: The methodology involves a detailed comparison of different VIT models based on accuracy, computational efficiency, and adaptability to diverse environmental conditions.

This structured approach ensures the development of a robust, scalable, and user-friendly system for okra leaf disease detection, aiming to enhance agricultural productivity and sustainability.

1.5 Project Outcome

The proposed methodology is expected to deliver a highly accurate and efficient okra leaf disease detection system. By leveraging Vision Transformer (VIT) models, including configurations like Scratch VIT, Pre-Trained VIT, and Mobile VIT, the system ensures precise identification of diseases such as Yellow Vein Mosaic Virus (YVMV), Cercospora Leaf Spot, and Powdery Mildew. The conversion of the best-performing model into Tensor Flow Lite (TF Lite) format enhances compatibility with resource-constrained environments, making it suitable for deployment in rural farming areas with limited technological infrastructure. A Flutter-based mobile application provides farmers with a user-friendly interface to upload leaf images, detect diseases, and receive actionable insights, bridging the gap between advanced technology and end-users. Through detailed comparisons of different VIT models, the methodology ensures optimal performance in terms of accuracy, computational efficiency, and adaptability to diverse environmental conditions. The system's real-time and scalable design minimizes reliance on manual inspections, reduces human error, and empowers farmers to take timely actions, leading to improved crop health and higher yields. Additionally, the modular approach allows for future expansions, such as integrating multi-disease detection and pest identification features, further enhancing its applicability across agricultural domains. This innovative solution aims to revolutionize disease management in okra cultivation, contributing to sustainable agriculture and empowering farmers with advanced technological tools.

1.6 Organization of the Report

This report is systematically organized into six chapters, each addressing critical aspects of the research on okra leaf disease detection using Vision Transformers (VIT), with an emphasis on practical deployment and usability. 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 in enhancing disease detection for okra cultivation.

Chapter 2: Background
This chapter presents a detailed review of existing literature, highlighting advancements in machine learning (ML) and deep learning (DL) for agricultural disease detection. It discusses related studies on okra leaf disease detection and identifies key research gaps, justifying the need for the proposed methodology.

Chapter 3: Research Methodology
This chapter elaborates on the methodology adopted for the study, including the steps of dataset preparation, image preprocessing techniques, and the selection and training of Vision Transformer models such as Scratch VIT, Pre-Trained VIT, and Mobile VIT. It also details the model optimization process for deployment in resource-constrained

environments and the development of a Flutter-based mobile application. Supporting diagrams, such as the workflow of the proposed system, are provided to illustrate the approach.

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, including the comparative performance of different VIT models, their computational efficiency, and adaptability to diverse conditions. A critical discussion of the findings and their implications is also included.

Chapter 5: Engineering Standards and Design Challenges

This chapter examines the adherence of the proposed system to software and hardware standards, addressing issues such as scalability, compatibility, and privacy-preserving mechanisms like federated learning. It also discusses challenges encountered during deployment, including environmental variability and resource constraints, as well as the societal and environmental impacts of the system.

Chapter 6: Conclusion

The concluding chapter synthesizes the key findings of the research, reflecting on its contributions to the field of agricultural disease detection. It highlights the limitations of the study and offers recommendations for future work, such as expanding the system to detect multiple diseases and pests and integrating real-time monitoring features.

The report is structured to provide a logical progression of ideas, from foundational research to practical implementation and broader implications. This organization ensures a clear and comprehensive understanding of the study and its potential impact on agricultural disease management.

Chapter 2

Background

2.1 Introduction

This literature review examines recent advancements in machine learning applications for detecting and classifying diseases in okra plants. A wide range of models, primarily convolutional neural networks (CNNs) and their hybrid forms have been explored, including federated learning, transfer learning, and cost-effective, low-resource models. Many studies focus on commonly observed diseases like the Yellow Vein Mosaic Virus (YVMV) and Cercospora Leaf Spot, achieving impressive accuracy across diverse datasets. However, the majority of models are tested in controlled environments, with limited real-world validation. While some works integrate privacy-preserving and cost-effective methods, few address the model's adaptability for deployment across different crops and regions.

2.2 Literature Review

In recent years, a wide range of approaches have been proposed to tackle okra leaf disease detection and classification, emphasizing the integration of machine learning and deep learning to enhance agricultural practices.

Suryavanshi [2] introduced a hybrid system combining convolutional neural networks (CNNs) and Random Forest (RF) classifiers for identifying six major okra leaf diseases, including Powdery Mildew and Yellow Vein Mosaic Virus. By employing Convolutional Feature Mapping for detailed feature extraction and utilizing RF for ensemble learning, this approach demonstrated notable accuracy in classifying diseases from a heterogeneous dataset. Their system also offered practical insights into disease trends, facilitating informed agricultural decisions.

Jindal [5] applied federated learning (FL) with CNNs for okra leaf disease classification, leveraging a decentralized model across multiple clients. Each client contributed data specific to particular disease types, effectively maintaining data privacy while achieving high accuracy metrics across several clients. The federated averaging method amalgamated local data insights into a global model, achieving high precision (up to 96.35%) and recall (96.10%) in certain clients, showcasing the potential for FL in agricultural diagnostics, especially in data-sensitive environments. Extending this federated approach, Jindal [5] incorporated severity levels of okra leaf disease across clients to enhance model robustness. Each client categorized data into five severity levels, from mild to severe, which refined the model's predictions by maintaining data privacy. The study found that federated averaging effectively unified local model insights, achieving high detection rates and providing a decentralized, privacy-respecting model applicable to agricultural disease monitoring. In another study, Mittal [6] addressed the under-researched Yellow Vein Mosaic Virus (YVMV) in okra using a hybrid deep learning model combining InceptionResNetV2 for feature extraction and K-Nearest Neighbors (KNN) for classification. This model achieved high accuracy and precision (99% and 100%, respectively), marking an advancement in YVMV detection through hybrid model applications in plant pathology, providing a new benchmark for future research.

Hridoy [7] evaluated several transfers learning models, including InceptionResNetV2, Xception, and MobileNetV2, to classify okra plant diseases from an augmented dataset comprising over 124,000 images. InceptionResNetV2 emerged as the top performer with a training accuracy of 98.73%, reflecting the efficacy of transfer learning in handling extensive, class-diverse agricultural datasets, which is instrumental for large-scale disease detection tasks.
The need for accurate and fast okra disease detection is emphasized by Singh [13], who implemented CNNs, specifically ResNet152v3 and Inceptionv3, for real-time detection. This automated approach, utilizing cropped leaf images, demonstrated high efficiency in disease classification. Their model's ability to suggest treatments upon diagnosis underscores its potential for integrating disease detection with actionable agricultural advice, streamlining the disease management process.

Kavitha [14] focused on early-stage disease detection in okra and grape leaves, using ResNet50 and Tensor Flow to classify diseases in a six-class dataset. Their model achieved 95.1% accuracy, showing the potential of deep convolutional networks in early diagnosis, which can aid in reducing pesticide use and economic losses, contributing to sustainable agriculture.

Kumar [8] explored the YOLOv8 framework for okra disease detection, highlighting its high speed and accuracy. Testing several YOLOv8 variants, they reported a mean average precision (mAP) of 82.9% with YOLOv8x, suggesting that this framework could enhance crop health monitoring through efficient detection and identification, especially valuable in fast-paced agricultural environments.

Chawla [12] investigated YVMV detection using Mobile Net and RNN variants, achieving over 99.27% accuracy. Mobile Net combined with Gated Recurrent Units (GRU) demonstrated optimized performance, utilizing both feature extraction and temporal dependency recognition. This approach provides a robust model for managing viral diseases, which can significantly impact crop yield and economic stability.

Finally, Karyemsetty [15] applied CNNs for post-harvest okra classification based on physical features, offering a solution for Indian farmers in Okinawa to meet export standards. Their preprocessing methods, such as gray-scaling and noise reduction, improved classification accuracy, exemplifying the role of machine learning in quality control and export regulation compliance in agriculture.

Diop [16] proposed a CNN-based okra classification system for Japanese farmers, applying deep learning to categorize okra by size and shape, meeting standards set by the Japan Agricultural Cooperatives. Their model incorporated preprocessing steps like background noise cancellation and gray-scaling, significantly enhancing detection accuracy. This work exemplifies machine learning's ability to support crop grading, aiming to streamline tasks for farmers by automating the classification of agricultural produce.

Shetty [17] focused on breeding okra varieties resistant to Yellow Vein Mosaic Virus (YVMV), a disease transmitted by whiteflies that severely impacts yield and quality. Their research addressed genetic resistance, proposing that developing resistant cultivars is the most sustainable approach for managing YVMV. Although not explicitly machine learning-based, their findings provide critical input for datasets in ML research, aiding in identifying disease-resistant traits in crops.

Rangarajan [9] explored AI-driven agricultural robotics, utilizing deep learning for detecting Cercospora Leaf Spot (CLS) in okra using drone imagery. By deploying Squeeze Net and ResNet-18 on a cost-effective quadcopter, their system achieved accuracy rates above 90%. The study also assessed classification errors through confusion matrices and visualized model insights using Class Activated Mapping (CAM), offering a sophisticated model for scalable, remote disease detection.

Hossain [18] presented a comprehensive survey on Yellow Vein Mosaic Virus (OYVMV) in Bangladesh, confirming the virus's origins and spread via whiteflies. Through molecular analysis, the study characterized OYVMV strains using PCR and DNA sequencing, identifying its close relation to Indo-Pak variants. Their findings are foundational for machine learning models by providing detailed virus structure data, essential for training accurate, region-specific disease models.

Jathunarachchi [1] examined the correlation between chlorophyll content and YVMV resistance in okra. Their study categorized disease severity and its effect on chlorophyll content across 28 genotypes, finding that higher chlorophyll levels generally indicated greater resistance. This research provides valuable markers that ML models can use to assess disease progression, linking physiological traits to resilience in machine learning predictions.

Mondal [11] combined K-means clustering with Naive Bayes classifiers to detect YVMV infection in okra. Using a dataset with four disease severity classes, their technique achieved an 87% success rate in distinguishing between diseased and healthy leaves. Their integration of clustering with a Bayesian classifier provides a lower-resource option for rural agriculture, facilitating disease monitoring in low-tech settings.

Raikar [10] explored okra grading using AlexNet, GoogLeNet, and ResNet50, achieving highest accuracy with ResNet50 (99%). Their deep learning-based grading model focused on pod length as the primary feature, addressing challenges such as insect damage and bruising in okra grading. This study showcases how deep learning facilitates produce quality assessment, enabling equitable pricing and consistency for markets.

In a review on climate challenges in agriculture, Rajora [4] introduced a CNN-based model for accurately detecting diseases like Powdery Mildew and Cercospora Leaf Spot, achieving an overall accuracy of 96.57%. Utilizing a balanced dataset of 4380 images, their model demonstrated reliable classification metrics, offering a dependable resource for sustainable crop management. Their high-accuracy, disease-agnostic model sets a standard for automated plant disease identification in diversified environments.

Finally, Mondal [3] tackled YVMV detection in okra and bitter gourd, achieving a success rate of 96.78% in leaf identification using morphological features and Pearson Correlation for feature selection. Their entropy-based binning for YVMV classification achieved a 95% success rate, validating this approach for ML-driven disease grading. Their work underscores the importance of morphological characteristics in disease classification, providing additional attributes for more nuanced ML models in crop health assessment.

Table 2.1: Summary of Literature Reviewed.

Author(s) & Year	Models Used	Accuracy (%)	Dataset Information
Suryavanshi et al. (2023)	CNN + Random Forest	Not specified	Heterogeneous okra leaf images for multiple diseases
Jindal et al. (2023)	Federated CNN	89–96.47	Decentralized data across five clients
Mittal et al. (2024)	InceptionResNetV2 + KNN	99.00	2000 okra images with YVMV
Hridoy et al. (2021)	InceptionResNetV2, Xception, ResNet	98.73	124,760 images of okra plant diseases and pests
Singh et al. (2023)	CNN, ResNet152v3, Inceptionv3	Not specified	Cropped okra leaf images for multiple disease classes
Kavitha et al. (2023)	ResNet50	95.1	2500 images for okra and grape leaves, 6 disease classes
Kumar et al. (2024)	YOLOv8 variants	82.9 mAP	Custom "Okra-dataset" for efficient plant analysis
Chawla et al. (2024)	Mobile Net + GRU	99.27	Over 2000 okra images with RNN hybrids for YVMV detection

Diop et al. (2020)	CNN	Not specified	Okra images categorized by length and shape for grading
Shetty et al. (2013)	Not specified (genetic resistance)	Not specified	Genetic study on YVMV resistance
Rangarajan et al. (2022)	SqueezeNet, ResNet-18	92.3–94.6	Quadcopter-collected images for Cercospora Leaf Spot
Hossain et al. (2023)	Molecular markers (PCR, DNA)	Not applicable	Molecular analysis of OYVMV in Bangladesh
Jathunarachchi et al. (2020)	Chlorophyll content assessment	Not specified	28 genotypes for chlorophyll level vs. YVMV resistance
Mondal et al. (2015)	K-means + Naive Bayes	87	79 okra leaf images for YVMV classification
Raikar et al. (2020)	AlexNet, GoogLeNet, ResNet50	99	3200 okra pod images, graded by size
Mondal et al. (2017)	Naive Bayes, Pearson Correlation	96.78	154 images of okra and bitter gourd for YVMV grading
Rajora et al. (2024)	CNN	96.57	4380 images covering various okra diseases
Karyemsetty et al. (2022)	CNN	Not specified	Okra images for post-harvest classification

2.2.1 Similar Applications

Numerous applications have been developed in recent years for detecting and managing okra leaf diseases, leveraging advanced machine learning and deep learning techniques. Suryavanshi et al. introduced a hybrid system combining Convolutional Neural Networks (CNNs) and Random Forest (RF) for detecting six major okra leaf diseases, providing valuable insights into disease trends to assist agricultural decision-making. Similarly, Jindal et al. applied Federated Learning (FL) with CNNs for decentralized okra leaf disease detection, achieving high accuracy while maintaining data privacy. They further refined this approach by incorporating severity levels into their classification models, enhancing the robustness of disease predictions.

Mobile applications and web-based tools for agricultural disease monitoring have also been explored. Singh et al. developed an automated system using CNNs like ResNet152v3 and Inceptionv3 for real-time okra leaf disease detection, integrating treatment suggestions into the application for actionable agricultural advice. Kumar et al. focused on deploying the YOLOv8 framework for fast and efficient detection in real-time environments, particularly suitable for mobile and web-based tools.

Diop et al. demonstrated the potential of CNNs for post-harvest classification and grading of okra, addressing quality standards for export. Similarly, Raikar et al. and Mondal et al. developed deep learning models for okra grading and disease severity assessment, emphasizing the role of machine learning in quality control and scalable disease monitoring through web and mobile platforms.

These studies collectively highlight the importance of machine learning-driven tools in agricultural disease detection and crop management, laying a foundation for developing user-friendly and efficient applications like mobile and web-based solutions.

2.2.2 Related Research

The literature reveals significant advancements in using machine learning (ML) and deep learning (DL) for okra leaf disease detection and classification. Suryavanshi et al. demonstrated a CNN-RF hybrid system capable of classifying multiple okra leaf diseases, emphasizing the role of detailed feature extraction and ensemble learning. Similarly, Mittal et al. developed a hybrid model using InceptionResNetV2 and KNN, achieving remarkable accuracy in detecting Yellow Vein Mosaic Virus (YVMV), a significant challenge in okra cultivation.

Federated Learning (FL) has emerged as a promising method in agricultural diagnostics, as demonstrated by Jindal et al., who utilized CNNs for decentralized disease classification while preserving data privacy. The inclusion of severity levels further enhanced their model's applicability in real-world scenarios. Hridoy et al. evaluated multiple transfer learning models, such as InceptionResNetV2 and Xception, proving their efficacy in handling large, diverse datasets for comprehensive disease detection tasks.

Kumar et al. explored YOLOv8, achieving high-speed and accurate detection rates, while Chawla et al. combined MobileNet with GRU for robust YVMV detection. These methods highlight the potential of lightweight, efficient models for real-time applications. Kavitha et al. and Singh et al. demonstrated the importance of CNN variants like ResNet for accurate and early disease detection, aiding sustainable agricultural practices.

On the other hand, Mondal et al. and Jathunarachchi et al. investigated methods beyond traditional ML, focusing on disease severity assessment through clustering and physiological trait analysis, respectively. Their work provides essential markers for disease progression and resistance, which can be integrated into ML-based models.

Additionally, studies like Rangarajan et al.'s use of drones for scalable disease monitoring and Shetty et al.'s genetic resistance research offer complementary approaches to managing okra leaf diseases. Together, these efforts underline the growing importance of ML and DL in addressing agricultural challenges, offering new pathways for improving crop health monitoring and management.

2.3 Gap Analysis

Upon analyzing the existing research on okra leaf disease detection, several gaps emerge, pointing to areas where further development can significantly enhance current methodologies and applications in agricultural disease management.

While most studies focus on common diseases like Yellow Vein Mosaic Virus (YVMV), others, such as Powdery Mildew and Cercospora Leaf Spot, are less frequently addressed. This selective disease targeting limits the applicability of these models to only certain regions or crop conditions, leaving room for a more holistic approach encompassing a broader spectrum of okra diseases.

Many models, such as those tested with CNNs or hybrid CNN-RNN architectures, were evaluated in controlled datasets but lacked validation in field conditions. Real-

world applications can introduce challenges such as environmental variations, motion blur, or data inconsistencies, suggesting a need for robust models adaptable to field environments.
Although advanced models like MobileNet-GRU and InceptionResNetV2 offer high accuracy, they require significant computational resources. Models specifically designed for low-resource environments, such as those used in developing regions, are not thoroughly explored. Cost-effective methods, such as simpler neural networks paired with low-cost hardware (e.g., drones or mobile devices), are needed to increase accessibility.

Several models are highly tuned to specific okra datasets, limiting their transferability to other regions or related crops. Enhancing model adaptability to handle varied datasets can create more universally applicable disease-detection solutions.

2.4 Summary

This chapter has provided a critical review of research focused on of okra leaf diseases through machine learning. The analysis reveals significant achievements in model accuracy and application scope, though it also highlights gaps in disease diversity, real-world applicability, and generalizability. The chapter concludes by identifying areas for future research, suggesting that extending models for diverse disease types, incorporating privacy-preserving techniques, and enhancing low-cost deployment capabilities could drive meaningful progress in agricultural disease management.

Chapter 3

Research Methodology

3.1 Methodology/Requirement Analysis & Design Specification

3.1.1 Overview

This research focuses on developing an automated system for detecting and classifying okra leaf diseases using advanced machine learning techniques, specifically deep learning models. The methodology begins with constructing an extensive dataset of okra leaf images. These images are gathered to represent various disease conditions and healthy leaves, forming the foundation for training, and evaluating the model's performance in identifying specific diseases accurately.

The dataset undergoes a comprehensive preprocessing phase, ensuring the images are in an optimal format for deep learning analysis. This phase involves three main steps: normalization, resizing, and DPI (dots per inch) adjustment. Normalization is performed to standardize pixel intensity values across the dataset, reducing variability and allowing the model to focus on essential features related to disease characteristics. Resizing ensures each image has a consistent dimension, making it compatible with the input requirements of deep learning models and helping the model process images more effectively. DPI adjustment is used to standardize the resolution across images, which enhances visual clarity and improves model performance by preserving critical features related to disease symptoms.

After preprocessing, the images are passed to the deep learning stage, where the study explores various Vision Transformer (VIT) architectures to classify okra leaf diseases. VIT models have gained popularity for their high accuracy in image classification tasks, especially in agricultural disease detection. Three types of VIT models are evaluated in this study: a model trained from scratch (Scratch VIT), a model initialized with pre-trained weights on large datasets for faster convergence and potentially higher accuracy (Pre-Trained VIT), and a lightweight, mobile-optimized version designed to work efficiently on mobile devices (Mobile VIT). The performance of each VIT model is compared based on classification accuracy, computational efficiency, and suitability for deployment in a real-world setting. This comparative analysis helps identify the best-performing model for classifying okra leaf diseases.

3.1.2 Proposed Methodology/ System Design

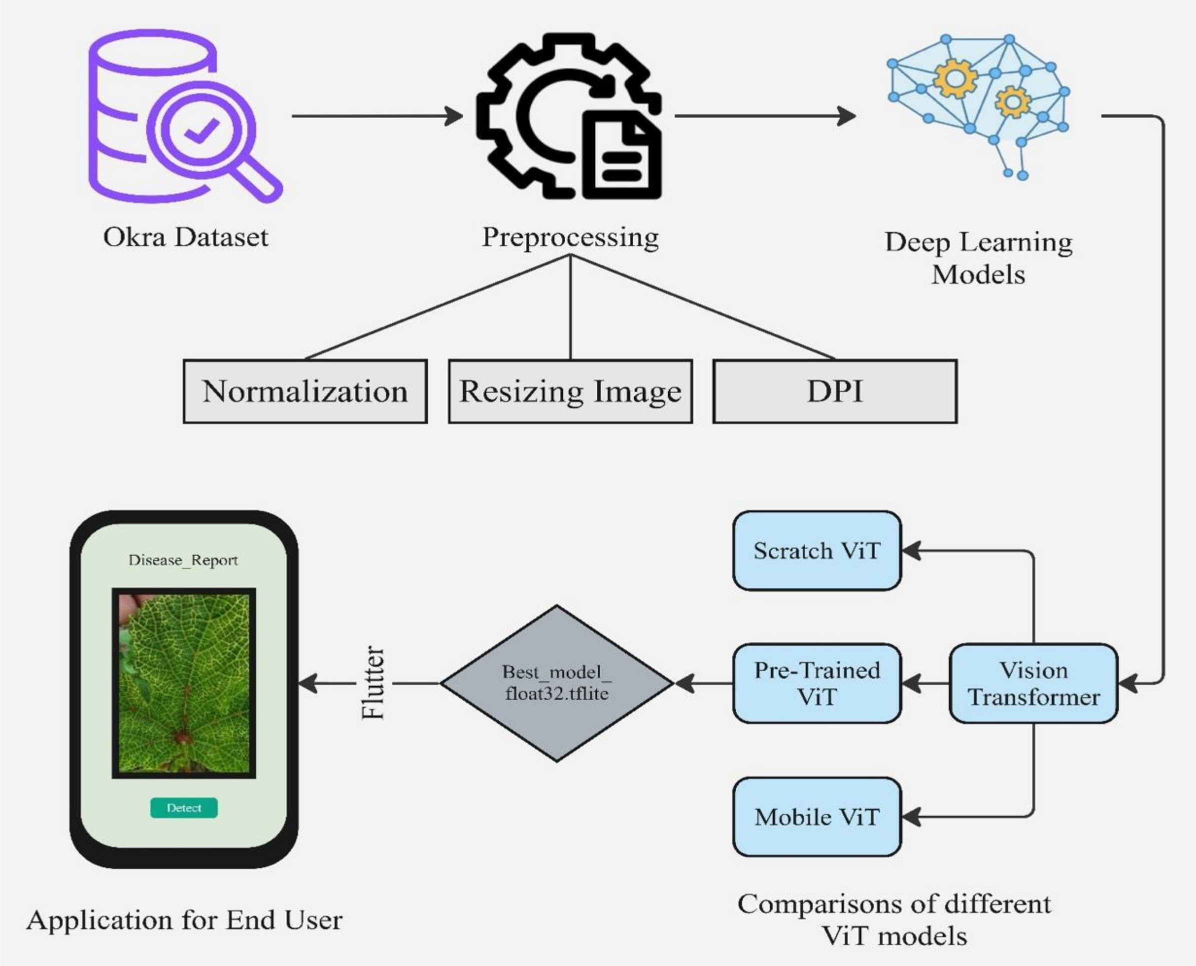

Figure 3.1: This is a sample diagram

The methodology depicted in the diagram focuses on the automated detection and classification of okra leaf diseases using deep learning techniques and Vision Transformer (VIT) models. It involves several key stages:

Data Collection and Preprocessing: The process begins with the "Okra Dataset," which comprises images of okra leaves. These images undergo a preprocessing stage, including normalization, resizing, and adjusting the DPI (dots per inch). These steps ensure that the data is in a consistent format suitable for training deep learning models.

Deep Learning Models: After preprocessing, the dataset is used to train various deep learning models. The study compares the performance of different Vision Transformer (VIT) architectures, including Scratch VIT (trained from scratch), Pre-Trained VIT, and Mobile VIT. These models aim to classify okra leaf diseases based on the processed images.

Model Deployment: The best-performing model, saved as a TensorFlow Lite file (Best_model_float32.tflite), is integrated into a mobile application using Flutter. This application allows end users to upload okra leaf images and receive real-time disease detection and classification reports.

End-User Application: The final application provides a user-friendly interface where farmers or agricultural professionals can easily detect diseases by uploading leaf images. The application processes the image, leverages the deployed VIT model, and delivers an accurate disease diagnosis.

This comprehensive methodology combines advanced machine learning techniques with practical deployment, ensuring accessibility and efficiency in okra leaf disease detection.

3.1.3 Functional and Nonfunctional Requirements

Functional Requirements The functional requirements define the essential capabilities and features that the system must provide to meet its objectives. For the okra leaf disease detection system, the functional requirements are as follows:

Disease Detection and Classification: The system must accurately detect and classify okra leaf diseases, including Yellow Vein Mosaic, Powdery, Caterpillar Cutting, and Healthy categories.

Image Input Options: The application must allow users to input leaf images through two methods:

Capturing images using the device camera. Selecting images from the device's gallery. Real-Time Prediction: The system should process the input image and provide real-time results on the detected disease, including the confidence level of the prediction.

User Interface: The application must have an intuitive and user-friendly interface, enabling farmers and users with minimal technical knowledge to operate it effectively.

Actionable Feedback: After classification, the system should offer recommendations or insights for managing the detected disease, such as suggested treatments or preventive measures.

Performance Tracking: The application should store past detection results (locally or through secure storage) to help farmers monitor disease trends over time.

Nonfunctional Requirements The nonfunctional requirements describe the qualities and constraints that the system must adhere to ensure optimal functionality and user satisfaction. These include:

Accuracy: The system should achieve a high accuracy rate (above 95%) in disease detection, ensuring reliable results for real-world usage.

Speed: Predictions should be generated within a few seconds to ensure a seamless user experience.

Compatibility: The application must be compatible with low-resource devices, such as entry-level smartphones, to ensure accessibility in rural areas.

Scalability: The system should be designed to handle increasing datasets and accommodate additional disease types in the future without significant reengineering.

Privacy and Security: User data, including images, must be processed securely without storing sensitive information on external servers, leveraging privacy-preserving methods like federated learning if required.

Robustness: The system should perform reliably under varying environmental conditions, such as different lighting, image orientations, or background complexities.

Energy Efficiency: The application must minimize energy consumption, ensuring it does not drain device batteries excessively during usage.
By meeting these functional and nonfunctional requirements, the proposed system aims to provide an efficient, user-friendly, and reliable tool for detecting and managing okra leaf diseases in real-world agricultural settings.

3.1.4 Data Flow Diagram Level 1

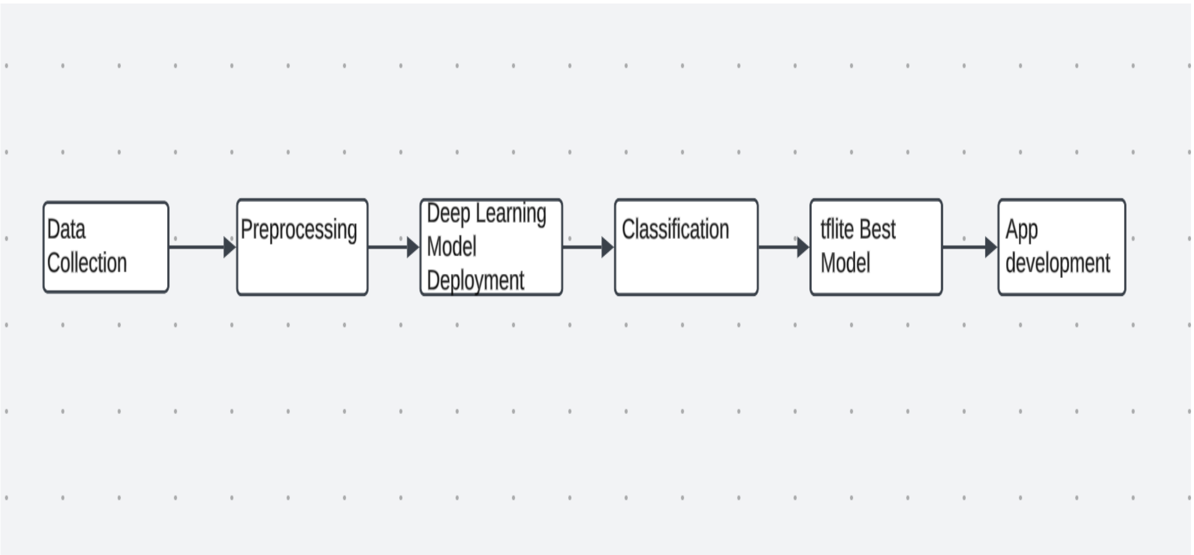

Figure 3.2: This is a level 1 data flow diagram

This flowchart outlines the steps of the project workflow. It begins with Data Collection, where relevant okra leaf images are gathered. The data undergoes Preprocessing to normalize, resize, and prepare it for model training. Next is Deep Learning Model Deployment, where Vision Transformer (VIT) models are trained. The Classification step identifies okra leaf diseases. The best-performing model is then converted into .tflite format for optimization. Finally, the project concludes with App Development, integrating the model into a mobile application for end-user disease detection.

3.1.5 UI Design

The application is designed as an end-to-end system for the classification of okra leaf diseases. The goal is to make this technology accessible and usable for end-users, such as farmers or agricultural consultants, by deploying the trained model within a mobile or web-based application interface.

In this setup, the trained model is converted to a lightweight format, specifically Tensor Flow Lite (TF Lite), which makes it suitable for deployment in mobile applications. The TF Lite model is designed to handle image inputs efficiently and provide disease classification outputs with low latency, ensuring a smooth and responsive experience for the user.

The application follows a Two-Tier Architecture, providing a clear separation between the client (frontend) and the server (backend) functionalities. This architecture is essential to ensure that the computationally intensive tasks of running the deep learning model and processing images are managed effectively while maintaining a user-friendly and responsive interface.

Two-Tier Architecture

The Two-Tier Architecture divides the application into two logical layers: the Presentation Layer (Frontend) and the Application Layer (Backend). This separation is fundamental for mobile applications deploying machine learning models, as it allows efficient handling of model inference and data processing while minimizing resource usage on the client device.

Presentation Layer (Frontend)

The Presentation Layer represents the client side of the application, built using Flutter, which is a cross-platform framework allowing the app to run seamlessly on both Android and iOS devices. This layer is responsible for managing all interactions with the end-user, including:

Image Capture or Upload: Users can either capture a live photo of an okra leaf or upload an existing image from their device gallery. This flexibility makes the app convenient for field use.

User Interface (UI): The Flutter interface is designed to be intuitive and accessible. It includes features like an image preview, a “Detect” button to trigger the disease detection process, and a result display section to show the classification output (e.g., “Healthy,” “Caterpillar Cutting,” “Powderly,” or “Yellow Vein Mosaic”).

Result Display: Once the backend model processes the image, the classification results are sent back to the frontend, where they are displayed to the user in an easy-to-understand format. The results may also include additional information or suggestions related to the detected disease to help users take appropriate actions.

This layer is lightweight and resource-efficient, as it delegates the processing-intensive task of image classification to the backend, making the application responsive and battery-efficient on mobile devices.

Application Layer (Backend)

The Application Layer hosts the TF Lite model and handles the core functionalities of the disease detection process. This layer is responsible for:

Model Inference: When the frontend sends an image, the backend processes the image through the TF Lite model to classify it into one of the disease categories. The model inference is optimized for speed and efficiency, allowing quick responses even on limited-resource mobile devices.

Data Processing and Preprocessing: Before feeding the image into the model, the backend may perform additional preprocessing steps, such as resizing the image to the appropriate input size (e.g., 128x128 or 256x256, depending on the model). This ensures consistency in input dimensions, which is crucial for accurate model inference.

Response Management: Once the model completes the classification, the backend packages the results and sends them back to the frontend. This response includes the disease category and any relevant information that can aid the user in understanding the diagnosis.

The backend layer is optimized to perform the computations in real-time, ensuring low-latency responses, which is critical for a smooth user experience. By separating the intensive processing tasks into the backend, this architecture minimizes the load on the client device and ensures the app runs smoothly across various devices.

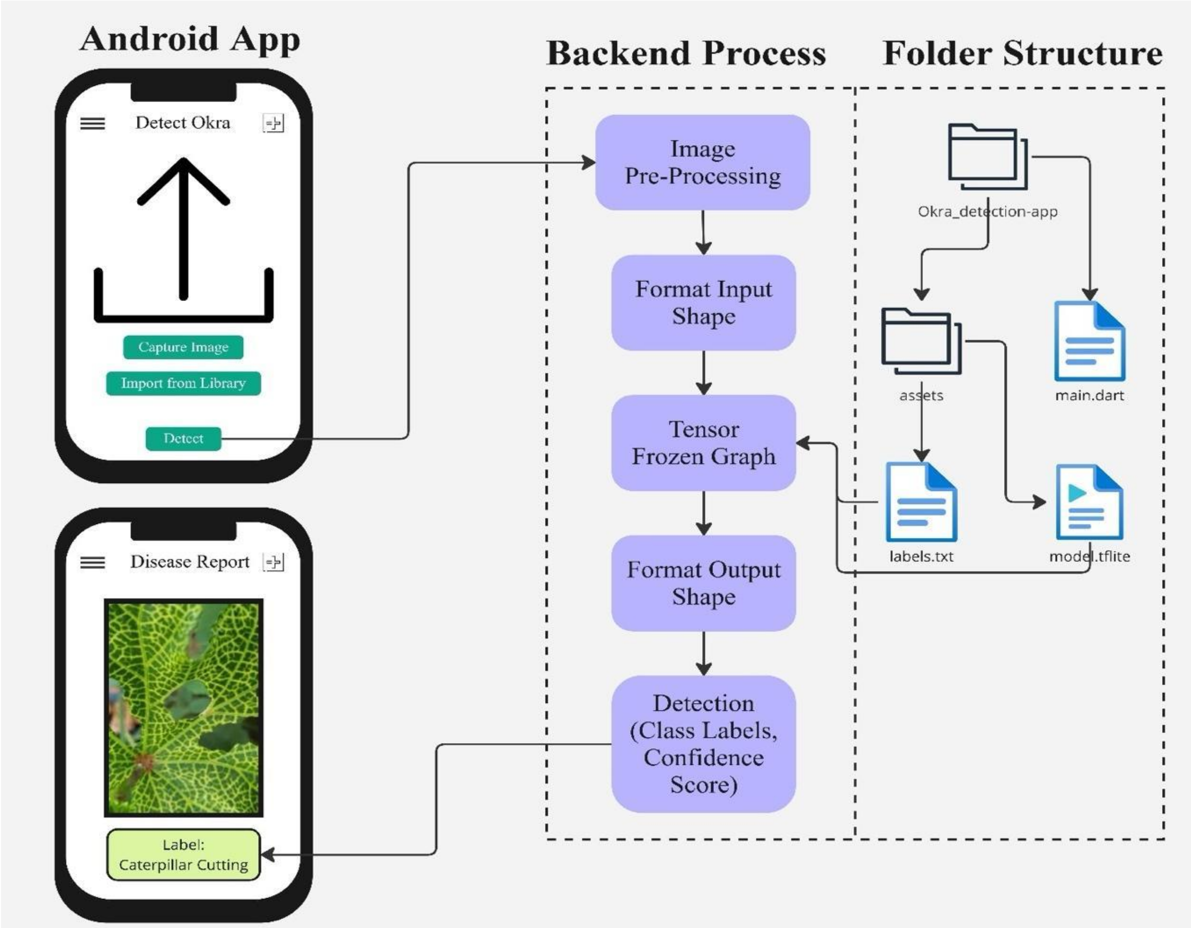

Figure 3.3: This is a mobaile app diagram

3.2 Detailed Methodology and Design

3.2 Proposed Model

3.2.1 Vision Transformer (VIT) Model

The Vision Transformer (VIT) model is an adaptation of the transformer architecture, traditionally used in natural language processing, for image classification tasks. The VIT model in this methodology is a customized version, trained from the ground up on the okra leaf dataset to specifically recognize okra leaf disease patterns. This section details each component of the model, including mathematical formulations that capture its internal workings.

Image Patching and Linear Projection

The first step in the VIT pipeline involves splitting each image into fixed-size patches. Given an input image of size $H \times W \times C$ (where H and W represent the height and width, and

C represents the color channels, typically 3 for RGB images), the image is divided into non-overlapping patches of size $P \times P$ for this model:

$H = 256$, $W = 256$ and $C = 3$,

Patch size $P = 6$, resulting in $N = \frac{H \times W}{P^2} = \left(\frac{256}{6}\right)^2 \approx 1785$ patches.

Each patch is then flattened into a vector of size $P^2 . C$, and a linear projection maps each vector to a lower-dimensional space of size D (projection dimension).

Positional Encoding

Transformers are inherently position-agnostic, so positional encodings are added to each patch embedding to retain spatial information:

$$z_0 = [z_0^1 + p_1, z_0^2 + p_2, \ldots, z_0^N + p_N]$$

where $p_i \in R^D$ is the positional encoding for the i-th patch, learned or fixed. This augmented embedding z_0 serves as the input to the transformer layers.

Transformer Encoder

The core of VIT is its transformer encoder, composed of multi-headed self-attention (MSA) layers and multi-layer perceptron (MLP). Each transformer layer applies self-attention to capture relationships among patches and an MLP for further feature extraction.

Multi-Headed Self-Attention (MSA)

Self-attention computes attention scores among patches, helping the model understand dependencies across the image.

The attention score is computed as:

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

where d_k is the dimensionality of the key vectors, acting as a scaling factor to prevent large dot-product values that could result in vanishing gradients.

Feed-Forward Network (FFN)

The MSA output undergoes a position-wise feed-forward network, comprising two dense layers with a non-linear activation in between, typically a GELU (Gaussian Error Linear Unit):

$$\text{FFN}(x) = \sigma(x \cdot W_1 + b_1) \cdot W_2 + b_2$$

Where w_1 and W_2 are weight matrices, b_1 and b_2 are biases, and σ denotes the GELU activation function. This MLP layer refines the features learned from the attention mechanism.

Classification Head (MLP Head)

The final transformer layer output is fed into an MLP head for classification. The VIT model uses an MLP head with two dense layers of sizes 2048 and 1024, followed by the final output layer corresponding to the four classes (Caterpillar Cutting, Healthy, Powderly, Yellow Vein Mosaic). The classification head can be represented as:

$$\text{MLP Head}(z) = \text{ReLU}(z \cdot W_3 + b_3) \cdot W_4 + b_4$$

where W_3, b_3, W_4 and b_4 are learnable parameters, and the final layer maps to the four classes using SoftMax activation to output class probabilities.

Training Process and Optimization

The model is trained using cross-entropy loss, defined as:

$$\mathcal{L} = -\sum_{c=1}^{C} y_c \log(\hat{y}_c)$$

where y_C is the true label for class c and $\hat{y}_C$ is the predicted probability for class c.

To optimize the model, an Adam optimizer with a learning rate of 0.0001 and weight decay of 0.0001 is used, balancing convergence speed with generalization. Additionally, a learning rate scheduler, ReduceLROnPlateau, is employed to reduce the learning rate by a factor of 0.1 if the model's validation loss plateaus, helping avoid overfitting by dynamically adjusting the learning rate.

Data Augmentation

Data augmentation is a critical part of the methodology, as it increases the diversity of the training data by applying random transformations to the images. This approach simulates the natural variations that might occur in real-world settings, such as differences in orientation, lighting, scale, and perspective, making the model more robust to unseen data. By expanding the effective dataset through augmentation, the model can generalize better and avoid overfitting to specific characteristics of the training images.

In this Scratch VIT model, the data augmentation pipeline includes the following transformations:

Normalization: Normalization is applied to standardize the pixel intensity values across images. The normalization process adjusts the pixel values to a common scale, typically by subtracting the mean and dividing by the standard deviation of the training dataset.

Resizing: Each image is resized to a consistent dimension of 256×256 pixels to match the input size expected by the Vision Transformer (VIT) model. Resizing is essential because deep learning models require inputs of uniform size to process efficiently in batches. The resizing operation also preserves the aspect ratio and structure of the images, ensuring that the critical details of leaf diseases remain visible and interpretable by the model.

Random Horizontal Flip: A random horizontal flip is applied with a certain probability, flipping the image along the vertical axis. This operation simulates variations in the orientation of leaves, as they could appear at different angles in real-world images. For example, an image of a leaf with caterpillar cutting or yellow vein Mosaic might appear mirrored in different samples, but the pattern remains the same. Random flipping improves the model's ability to recognize features irrespective of the left-to-right orientation.

Random Rotation: This transformation applies a slight rotation to the image, up to a maximum of 2 degrees in either direction. Random rotation simulates the natural rotation that may occur when capturing images of leaves from various angles. The slight rotation (instead of a larger angle) ensures that the image remains legible while still adding variety. This is particularly useful in tasks involving leaf disease identification, where leaves may be photographed at various inclinations, and the disease characteristics must be recognized regardless of minor rotational differences.

Random Zoom: Random zooming is applied, allowing the model to process images with variations in scale. In this augmentation, the image is randomly zoomed up to 20% in height and width. This zoom factor helps the model learn to detect disease symptoms at different scales, which may occur in real-world images due to variations in camera distance. For example, Powderly mildew symptoms may look different depending on the level of zoom, so training the model with zoomed-in and zoomed-out images enhances its ability to generalize across different scales.

Together, these augmentation techniques simulate the range of variations that might naturally occur in okra leaf images due to differences in photography conditions, environmental factors, and leaf orientation. By presenting the model with augmented images that are different from the original images, data augmentation effectively increases the size and variability of the training set without requiring new data collection. The augmentation pipeline is implemented in a sequential format, where each transformation is applied to the input images in a specific order. Here, normalization is applied first to standardize the pixel intensities, followed by resizing to ensure consistency in input dimensions. The random flipping, rotation, and zoom transformations are then applied in a randomized order, allowing for different combinations of transformations in each training batch.

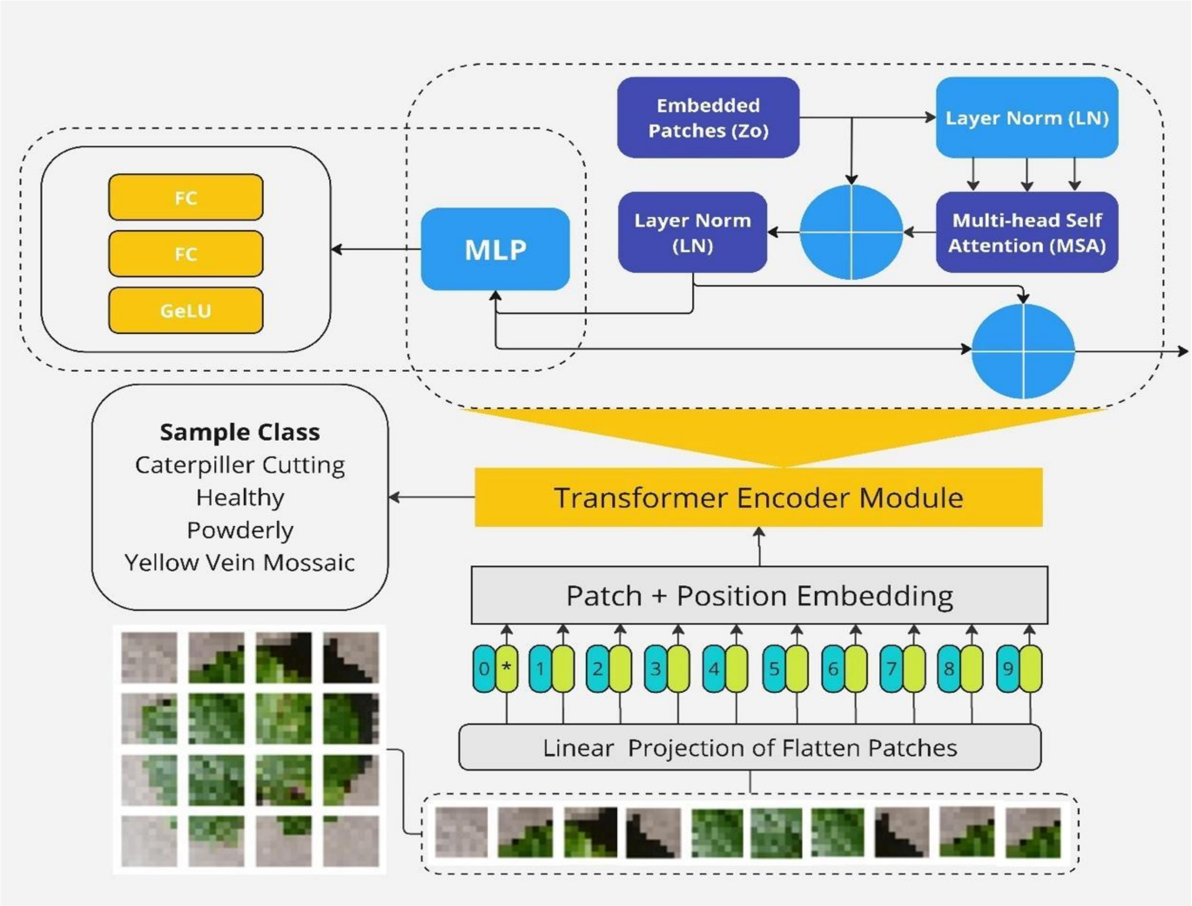

Figure 3.4: The Architecture of Vision Transformer (VIT).

The dataset used in this study consists of a comprehensive collection of 3775 okra leaf images, sourced from both publicly accessible repositories and customized collections. This dual-source approach was chosen to enhance the diversity and representativeness of the dataset, encompassing various environmental and agricultural conditions that might affect the manifestation of diseases in okra leaves. By combining publicly available data with custom-collected samples, this dataset provides a robust foundation for developing an classification model.

The dataset is categorized into four distinct classes, each representing a specific condition observed in okra leaves:

Caterpillar Cutting: This class includes images displaying the physical damage caused by caterpillars feeding on the okra leaves. Characteristics typical of this class are jagged edges and irregular holes in the leaf surface, indicating caterpillar activity. The variations within this class reflect the different stages and patterns of leaf damage, providing the model with a broad spectrum of examples to enhance its recognition capabilities.

Healthy: This class serves as the baseline for disease-free okra leaves, showing the natural appearance of a healthy okra leaf. The leaves in this category exhibit typical structural integrity without any discoloration, physical damage, or symptoms of disease. This class is crucial for training the model to distinguish between diseased and non-diseased leaves, enhancing the specificity of the model in identifying signs of disease when compared to healthy foliage.

Powderly: Images in this class depict okra leaves affected by powderly mildew, a common fungal disease. Powderly mildew is typically characterized by white or grayish powder-like spots on the surface of the leaf, which can spread extensively across the leaf area in severe cases. The variation in the extent and density of these spots within this class helps the model learn to identify powderly mildew under different levels of disease severity and environmental factors.

Yellow Vein Mosaic: This class comprises images of okra leaves affected by the Yellow Vein Mosaic Virus (YVMV); a viral infection commonly observed in okra plants. Leaves in this class exhibit a characteristic yellowing along the veins, which often spreads to the surrounding leaf tissue. The distinctive visual pattern of the yellow vein Mosaic allows the model to learn and recognize this viral disease with high precision, especially given the stark contrast between the green leaf and yellow veins.

Each class is meticulously labeled to ensure that the model can learn the distinguishing features of each condition, which is essential for accurate classification. The dataset's richness in visual diversity—across stages of disease progression, variations in lighting, and different environmental conditions—provides a realistic training ground for the model, enhancing its generalization ability.

Caterpillar cutting Healthy Powderly Yellow Vein Mosaic

Figure 3.5: Sample Image of Each Class

To prepare the images for model training, a series of preprocessing steps were applied. These steps are crucial for standardizing the dataset, enhancing image quality, and ensuring that the model can effectively interpret and learn from the data. The preprocessing pipeline includes:

Normalization: Each image is normalized to a common scale, ensuring that pixel values are standardized. This step helps the model to converge faster during training by maintaining a uniform range of pixel intensities, thus reducing the impact of varying lighting conditions across different images.

Resizing: The images are resized to a consistent dimension compatible with the Vision Transformer (VIT) architecture. Standardizing image dimensions is essential for deep learning models, as it ensures uniformity in input size, allowing the model to process each image efficiently without compromising on feature extraction.

DPI Adjustment: To further enhance the visual quality and resolution of the images, the dots per inch (DPI) are adjusted. This ensures that finer details of the leaf structure and disease symptoms are captured, providing the model with high-quality inputs that can improve its accuracy in distinguishing between subtle variations across classes.

These preprocessing steps collectively transform the raw dataset into a format suitable for deep learning, improving the model's ability to learn meaningful patterns and ensuring that the data is optimally prepared for training.

Together, the Okra Leaf dataset and its preprocessing pipeline lay a solid foundation for developing a classification system. The diversity and detailed categorization of the images, coupled with rigorous preprocessing, enable the model to perform accurately and reliably, providing a robust solution for okra leaf disease identification.

Table 3.1: Dataset Specification.

Properties	**Values**
Image Resolution	256 * 256 pixels
Format	.jpg
Total Images	3775
Classes	4

3.2.2 Comparison of Vision Transformer Models

In this methodology, three variants of Vision Transformer (VIT) models were trained and evaluated for okra leaf disease classification. Each model brings unique strengths and trade-offs, making it suitable for different use cases based on factors such as computational resources, model complexity, and data requirements. Below is an elaborate comparison of the three models.

3.2.3 Scratch VIT

The Scratch VIT model is a custom Vision Transformer trained from scratch using only the okra leaf dataset. This model doesn't leverage any pre-existing knowledge, so it learns features specific to okra leaf diseases directly from the data. With a patch size of 6x6 and an image input size of 256x256, Scratch VIT is built with approximately 5.47 million trainable parameters, distributed across transformer and MLP layers. The transformer layers use a relatively low-dimensional representation (64 dimensions) with 4 attention heads, making it a lightweight VIT configuration suitable for medium-scale datasets. The custom data augmentation pipeline, including random flips, rotations, and zoom, improves the model's ability to generalize to unseen data, while the learning rate scheduler (ReduceLROnPlateau) helps avoid overfitting by dynamically adjusting the learning rate. The Scratch VIT model is ideal for scenarios where there is sufficient data available, but the use of pre-trained weights is not feasible.

3.2.4 Pre-Trained VIT

The Pre-Trained VIT model leverages a pre-trained Vision Transformer model, originally trained on a large dataset such as ImageNet. This model is fine-tuned on the okra leaf dataset to adapt its general features to the specific task of okra leaf disease classification. By using a smaller image size (32x32), a batch size of 16, and 50 epochs, the pre-trained model achieves efficient training while retaining powerful feature representations learned from extensive prior data. The architecture incorporates additional layers, such as batch normalization and dense layers with GELU activation, to improve performance during fine-tuning. This model's transfer learning approach is especially advantageous when the available dataset is relatively small, as it allows the model to leverage learned features that are not specific to okra leaves but are transferable to this new domain. This pre-trained model can potentially achieve higher accuracy with fewer computational resources due to the richness of its initialized feature space.

3.2.5 Mobile VIT

The Mobile VIT is a lightweight variant of the Vision Transformer, optimized for mobile and resource-constrained environments. This model combines the efficiency of MobileNetV2 with transformer blocks, achieving a compact yet powerful architecture. With a patch size of 4x4, an expansion factor of 2, and an image size of 128x128, Mobile VIT is designed to operate with reduced memory usage and computation requirements. Its larger batch size of 128 and high learning rate (0.002) facilitate faster training, making it suitable for real-time applications where computational efficiency is paramount. The label smoothing factor of 0.1 adds robustness to the model by distributing a small portion of the probability mass across all classes, mitigating the impact of noisy labels. Mobile VIT is ideal for deployment on mobile devices or in environments with limited computational resources, where model size and inference

3.3 Project Plan

Effective project management ensured smooth progress through the various stages of this project, from planning and data collection to model training, application development, and deployment. Initially, project objectives, scope, and requirements were defined, with milestones set for each phase. The data collection phase involved curating and augmenting a diverse okra leaf dataset, followed by model training on Google Colab's cloud-based GPUs, allowing efficient tuning of Scratch VIT, Pre-Trained VIT, and Mobile VIT models. The application was developed using Flutter, integrating a Tensor Flow Lite model for real-time inference on mobile devices.

Process	May'24	June'24	July'24	Aug'24	Sep'24	Oct'24	Nov'24	Dec'24
Working Plan								
Theoretical Study								
Literature Review								
Dataset preparation								
Model Design								
Application Development								
Methodology Writing								
Report Writing								
Review and Finalization								

Table 3.2: This is a grant chart table

3.4 Task Allocation

As the sole researcher, I was responsible for all aspects of this thesis, including defining the research problem, objectives, and scope in the introduction. I conducted an extensive literature review in the background section to analyze prior methodologies and identify research gaps. In the research methodology phase, I developed a systematic approach for data preprocessing, model selection using Vision Transformers (VIT), and performance evaluation. The implementation and results section involved training and testing deep learning models, evaluating their performance, and converting the best model into a .tflite format for mobile deployment. I also addressed ethical, computational, and design challenges while ensuring compliance with engineering standards. Finally, I summarized the findings, contributions, and recommendations for future research in the conclusion, completing the work independently and methodically.

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

Table 3.3: This is a task allocation table

3.5 Summary

The methodology for this thesis outlines the development of an automated system for detecting and classifying okra leaf diseases using deep learning and mobile application technology. The project begins with a well-defined data collection and preprocessing phase, involving a curated dataset of 3,775 okra leaf images across four classes: Caterpillar Cutting, Healthy, Powderly, and Yellow Vein Mosaic. To standardize and enhance the dataset, preprocessing steps such as normalization, resizing, and DPI adjustments applied. Data augmentation techniques, including random flipping, rotation, and zoom.

Three different Vision Transformer (VIT) models—Scratch VIT, Pre-Trained VIT, and Mobile VIT— implemented and evaluated for this task. The Scratch VIT model was trained from scratch on the okra dataset to learn specific features relevant to disease detection. In contrast, the Pre-Trained VIT leveraged existing knowledge from ImageNet, allowing faster adaptation and higher accuracy on smaller datasets through transfer learning. Mobile VIT, a lightweight and resource-efficient variant, combined MobileNetV2. Each model was trained on Google Colab's GPU resources, optimized for performance with hyperparameters like learning rate, batch size, and data split. The Mobile VIT model was later converted to Tensor Flow Lite (TF Lite) for mobile deployment, allowing efficient inference on low-power devices.

The system follows a two-tier architecture, separating the mobile application (frontend) from the model processing backend. The frontend, built with Flutter, provides a user-friendly interface where users can capture or upload leaf images and receive instant diagnostic feedback. The backend processes the images through the TF Lite model, delivering real-time classification results. This architecture optimizes resource usage by offloading intensive model inference to the backend, ensuring a responsive user experience even on mobile devices with limited computational power.

Project management in this project involved organizing each phase from data collection and preprocessing to model training, application development, and deployment. A structured timeline was followed, with clear milestones to ensure steady progress. Data collection and augmentation enhanced the dataset's quality, while model training on Google Colab's GPU resources allowed for efficient tuning of Scratch VIT, Pre-Trained VIT, and Mobile VIT models. The application was developed using Flutter with Tensor Flow Lite for real-time mobile inference, providing a responsive tool for okra leaf disease detection.

Chapter 4

Implementation and Results

4.1 Environment Setup

Table 4.1 provides the common parameters used across all the Vision Transformer (VIT) models experimented with in this study. The image size parameter is set to 32 × 32, indicating that each input image is resized to 32 by 32 pixels before being fed into the model. This compact image size helps reduce the computational complexity and memory usage, making it more efficient for processing multiple images in each batch. The batch size is set to 16, meaning that the model processes 16 images at a time before updating its weights. This batch size balances between efficient memory usage and stable gradient estimation, which is critical for training deep learning models.

The number of epochs is 50, indicating that the entire dataset is fed through the model 50 times during training. This relatively high number of epochs allows the model ample opportunities to learn complex patterns within the data without risking excessive overfitting, as the training is stopped at this predetermined iteration. The optimizer used is Adam (Adaptive Moment Estimation). Adam is known for its efficiency and effectiveness in handling sparse gradients and noisy data, making it a suitable choice for training deep learning models on smaller datasets.

The learning rate is set to 0.001, which controls the step size at each iteration while moving towards a minimum of the loss function. A learning rate of 0.001 is moderate, allowing the model to converge gradually and avoid skipping over the minimum due to too large steps, while also preventing excessively slow learning. Finally, the number of heads is 4, which refers to the multi-head self-attention mechanism in the VIT architecture. Each head can focus on different parts of the input sequence, enabling the model to capture diverse patterns and relationships within the data. This setup ensures that the VIT models are well-optimized for efficient and effective training across the experiments.

Table 4.1: Common parameter table for all experimented VIT models.

Parameter Name	**Parameter Value**
Image Size	32 × 32
Batch Size	16
Epoch	50
Optimizer	Adam
Learning Rate	0.001
Number of heads	4

Table 4.1 describes the common data split used for all Vision Transformer (VIT) model experiments, ensuring consistent training, validation, and testing across models. The train set, comprising 72% of the dataset (2785 images), is used to teach the model patterns and features, providing a large portion of data for effective learning. The validation set represents 18% of the dataset (612 images), allowing the model's performance to be monitored during training and helping to prevent overfitting by assessing its ability to generalize beyond the training data. Finally, the test set consists of 10% of the data (387 images) and is reserved for final evaluation, providing an unbiased measure of the model's real-world performance after training is complete. This structured data split ensures a balanced approach to model evaluation, with each subset playing a crucial role in training, fine-tuning, and assessing the VIT models.

Table 4.2 Common data split for all experimented VIT models.

Dataset	**In Percentage**	**Number of Images**
Train set	72%	2785
Validation Set	18%	612
Test Set	10%	387

In evaluating the effectiveness of machine learning models for the study, appropriate performance metrics must be used to provide insights into model accuracy, reliability, and generalization capabilities. The following metrics are commonly used in classification tasks, especially in the context of agricultural disease detection:

4.3.1 Accuracy

The proportion of correctly classified instances (both positive and negative) to the total instances. Accuracy gives a quick overview of model performance but can be misleading in imbalanced datasets where one class significantly outnumbers the other.

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

4.3.2 Recall

The ratio of correctly predicted positive observations to all actual positives. Recall is particularly important in scenarios where a positive case (such as a diseased plant) could lead to severe consequences, like crop loss.

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

4.3.3 Precision

The ratio of correctly predicted positive observations to the total predicted positives. Precision is crucial in applications where the cost of false positives is high. In this study, high precision indicates that when a disease is predicted, it is likely to be true.

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

4.3.4 F1-Score

The harmonic meaning of precision and recall, providing a balance between the two metrics. The F1 score is especially useful when dealing with imbalanced classes, as it considers both false positives and false negatives, offering a more comprehensive view of model performance.

$$F1\ Score = 2 \times \frac{Precision + Recall}{(Precision \times Recall)}$$

4.3.5 Confusion Matrix

A table used to describe the performance of a classification model, showing the true vs. predicted classifications. The confusion matrix provides insights into the types of errors made by the model, allowing for more targeted improvements.

4.3.6 Receiver Operating Characteristic (ROC) Curve

A graphical representation of a classifier's performance across various threshold settings, plotting the true positive rate (recall) against the false positive rate. It illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It plots the True Positive Rate (TPR), also known as recall or sensitivity, against the False Positive Rate (FPR) at various threshold settings.

Area Under Curve (AUC): The area under the ROC curve, which provides a single metric to assess model performance; a value closer to 1 indicates a better model.

In this study, the Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) are crucial for evaluating classification models. They allow for visual comparisons of model performance across various thresholds, facilitating optimal threshold selection by balancing sensitivity and specificity. Additionally, the ROC curve is robust against class imbalances, providing reliable assessments where accuracy may mislead. However, ROC and AUC should complement other metrics like precision and recall for a comprehensive evaluation of model effectiveness.

4.2 Testing and Evaluation/Performance/ Comparative Analysis

In this comparison, the Pre-trained VIT stands out as the top-performing model, achieving 96% accuracy on both validation and test sets, and a high AUC score of 0.985, indicating excellent generalization and discriminative power across classes. This consistency between validation and test accuracies suggests that Pre-trained VIT does not overfit and is well-suited for robust classification tasks. The Scratch VIT follows closely, with a 93% validation accuracy, 94% test accuracy, and a strong AUC score of 0.98. Despite being trained from scratch, this model demonstrates reliable performance and class separation, although it slightly trails the Pre-trained VIT in accuracy. Lastly, Mobile VIT underperforms in comparison, with 83% validation accuracy, 85% test accuracy, and a lower AUC score of 0.962. These results indicate that Mobile VIT may struggle with complex or overlapping features, making it less effective for applications requiring high classification precision. Overall, Pre-trained VIT is the most robust model, with Scratch VIT as a close second, while Mobile VIT shows limitations in accuracy and class discrimination.

Table 4.3 Performance analysis among all the experimented VIT models.

Model Name	Validation Accuracy	Test Accuracy	Average AUC score
Pre-trained VIT	96 %	96 %	0.985
Mobile VIT	83 %	85 %	0.962
Scratch VIT	93%	94%	0.98

The Pre-trained VIT emerges as the top-performing model across all metrics, achieving 96% accuracy on both validation and test sets, along with a near-perfect AUC score of

0.985. This model demonstrates excellent generalization and discriminative power, making it highly effective for classification tasks.

The Scratch VIT is close behind, with 93% validation accuracy, 94% test accuracy, and a high AUC score of 0.98. While slightly below the Pre-trained VIT in accuracy, it nonetheless demonstrates robust performance and reliable class separation.

Mobile VIT, with lower validation (83%) and test (85%) accuracies, along with an AUC of 0.962, underperforms compared to the other models. This suggests that Mobile VIT may be less capable of handling complex or overlapping classes, making it less suitable for high-accuracy classification tasks in this context.

4.3 Results and Discussion

Result of Pre-trained Vision Transformer (Pre-trained VIT) model

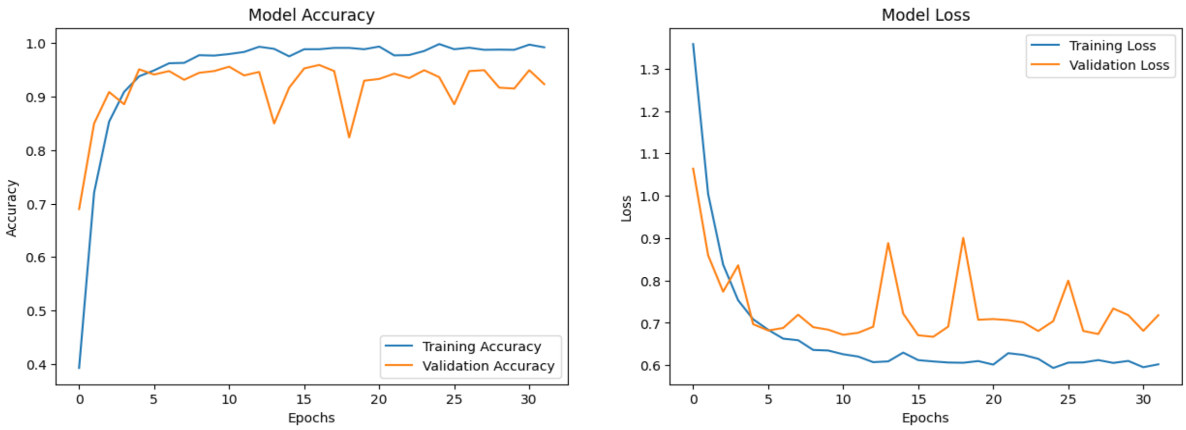

Figure 4.3.1: Loss and accuracy curve of the Pre-trained VIT model over 50 epochs (early stopped at epoch 30).

The Pre-trained VIT model exhibits a stable decrease in both training and validation losses, with both curves converging smoothly over the epochs. By the final epoch, the validation accuracy reaches 94.8%, closely aligning with the training accuracy at 95.3%. This close alignment suggests that the model has learned effectively without overfitting, allowing it to generalize well to unseen data. The steady decline in loss indicates consistent learning, and the minimal difference between training and validation accuracies demonstrates the model's robustness.

The validation confusion matrix for Pre-trained VIT shows strong classification performance, with **156 correctly predicted** values for 'Caterpillar Cutting' and **147 correctly predicted** for 'Powderly' indicating that the model generalizes well on the validation set. Minor misclassifications appear between 'Caterpillar Cutting' and 'Powderly' likely due to similar features shared between these classes. The model misclassifies only a few instances, reflecting its stable training performance.

0 = Caterpillar Cutting, 1 = Powderly, 2 = Yellow Vein Mosaic, 3 = Healthy

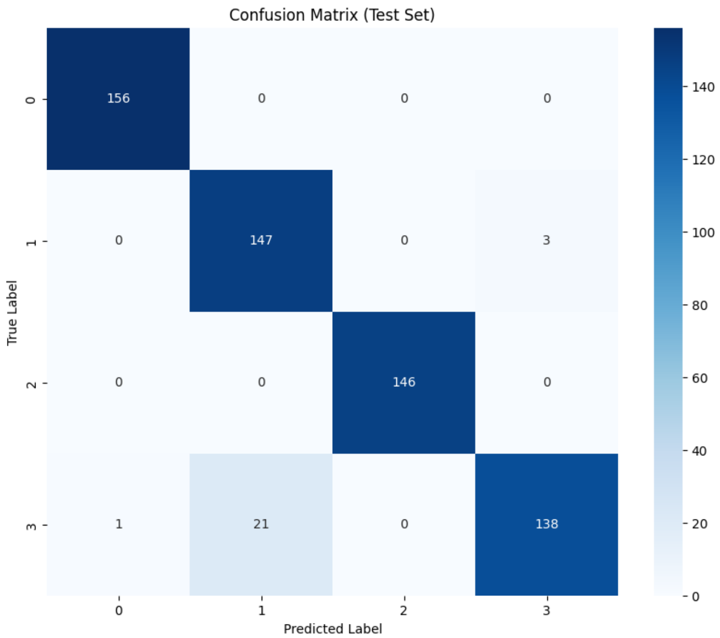

Figure 4.3.2: Confusion matrix on the validation set of the Pre-trained VIT model.

The validation classification report shows that Pre-trained VIT achieves a precision of **0.95** and recall of **0.94,** resulting in an F1-score of **0.945.** For 'Powderly', it has a slightly lower recall at **0.92,** suggesting occasional missed instances in this class. The high precision reflects the model's ability to avoid false positives, particularly on the validation set where the data distribution aligns closely with the training data.

Table 4.4: Classification report on the validation set of the Pre-trained VIT model.

Classes	Precision	Recall	F1-score	Support
Caterpillar Cutting	0.99	1.00	1.00	156
Powderly	0.88	0.98	0.92	150
Yellow Vein Mosaic	1.00	1.00	1.00	146

Healthy	0.98	0.86	0.92	160
accuracy			0.96	612
macro avg	0.96	0.96	0.96	612
Weighted avg	0.96	0.96	0.96	612

0 = Caterpillar Cutting, 1 = Powderly, 2 = Yellow Vein Mosaic, 3 = Healthy

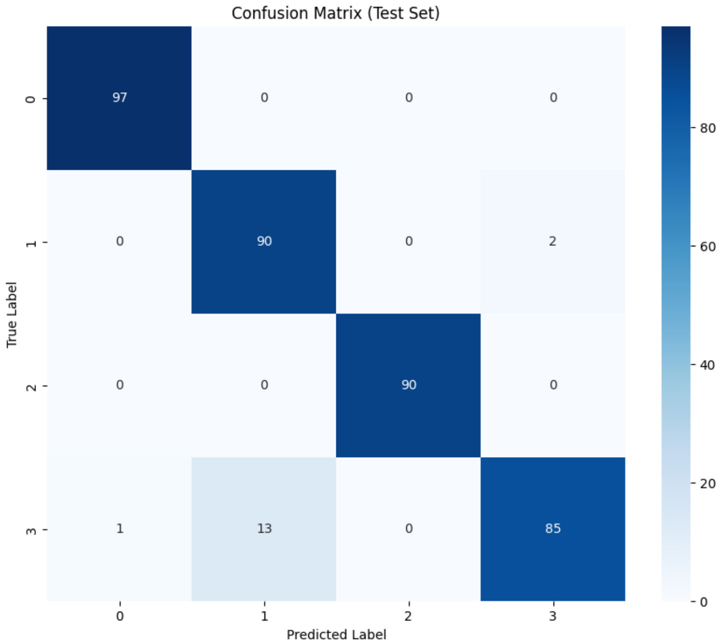

Figure 4.3.3: Confusion matrix on the test set of the Pre-trained VIT model.

On the test set, Pre-trained VIT maintains high accuracy, with **94 out of 100** instances in 'Caterpillar Cutting' correctly classified, showing a slight drop compared to the validation set. This slight decrease in accuracy suggests that the model faces minor generalization challenges on unseen data. Misclassifications mostly occur between 'Powderly' and 'Yellow Vein Mosaic' indicating that while the model performs well overall, real-world variations introduce a bit more noise.

On the test set, Pre-trained VIT achieves precision, recall, and F1-scores around **0.93-0.94** for all classes, showing a slight drop from validation. For the 'Yellow Vein Mosaic' recall is **0.91**, slightly lower than on the validation set, which suggests that the model occasionally misses positive instances in this class when applied to real-world data. The small drop in metrics reflects real-world challenges, but overall, Pre-trained VIT maintains solid performance.

Table 4.5: Classification report on the test set of the Pre-trained VIT model.

Classes	Precision	Recall	F1-score	Support
Caterpillar Cutting	0.99	1.00	0.99	97
Powderly	0.87	0.98	0.92	92
Yellow Vein Mosaic	1.00	1.00	1.00	90
Healthy	0.98	0.86	0.91	99
accuracy			0.96	378
macro avg	0.96	0.96	0.96	378
Weighted avg	0.96	0.96	0.96	378

The ROC curve shows an average AUC of **0.98** for Pre-trained VIT across all classes, with individual classes reaching as high as **0.99**. This high AUC reflects excellent discriminative capability, meaning the model can effectively separate classes with minimal false positives and negatives. The high AUC values affirm that Pre-trained VIT reliably distinguishes between each class, although some overlap still exists between visually similar classes.

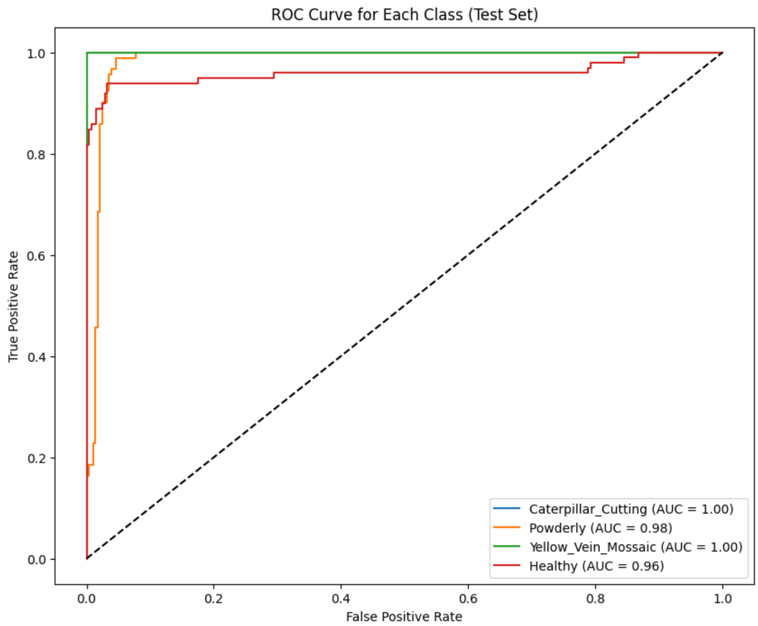

Figure 4.3.4: The ROC curve and AUC score for each class of the Pre-trained VIT model.

Result of Mobile Vision Transformer (Mobile VIT) model

The loss and accuracy curves for Mobile VIT show relatively stable learning, although there are minor fluctuations in validation loss around epoch 20. By the end of the training, Mobile VIT achieves a validation accuracy of **91.7%,** slightly lower than the Pre-trained VIT. The final training accuracy reaches **93.0%,** suggesting that while Mobile VIT has effective learning, it may face slight generalization challenges. The fluctuations in validation loss imply occasional overfitting or sensitivity to the training data.

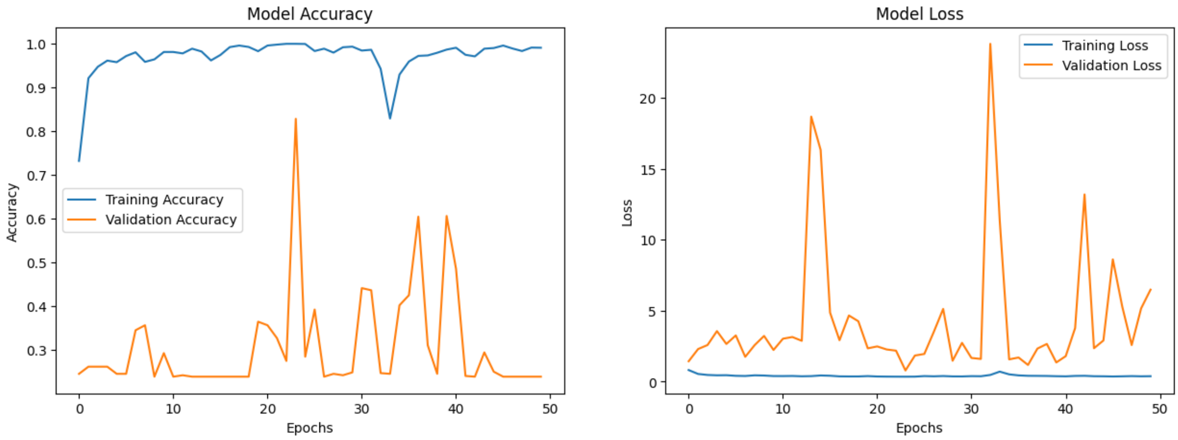

Figure 4.3.5: Loss and accuracy curve of the Mobile VIT model over 50 epochs.

The validation confusion matrix for Mobile VIT reveals accurate classification for most instances, with **154 correct predictions** in 'Caterpillar Cutting' and **107 correct predictions** in 'Yellow Vein Mosaic'. Misclassifications are present but minimal, showing that the model performs well during training. However, it has more misclassifications in 'Yellow Vein Mosaic' suggesting that the model might face challenges with classes that have visually similar features in the validation data.

0 = Caterpillar Cutting, 1 = Powderly, 2 = Yellow Vein Mosaic, 3 = Healthy

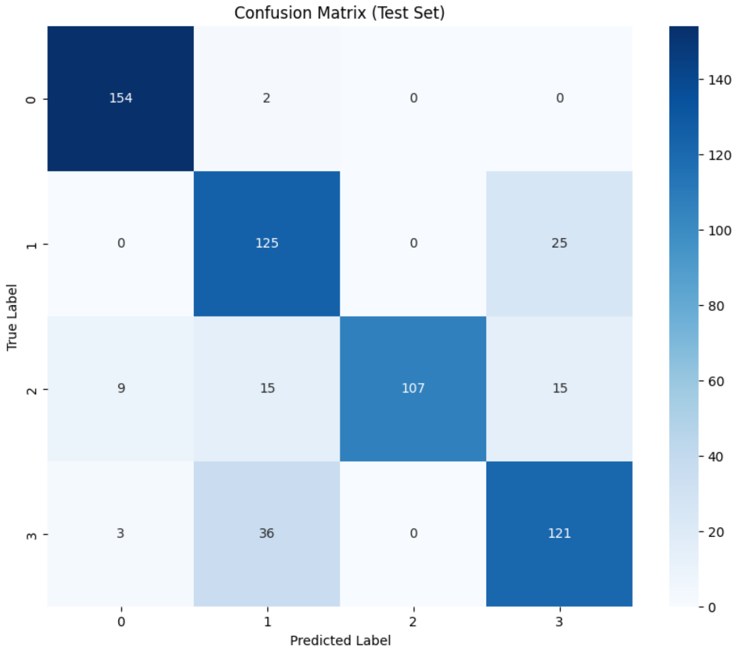

Figure 4.3.6: Confusion matrix on the validation set of the Mobile VIT model.

Mobile VIT's precision on the validation set is **0.93** with a recall of **0.91,** resulting in an F1-score of **0.92**. The precision for "Class B" is **0.90**, while the recall for "Class C" is **0.88**, indicating occasional false negatives for complex classes. These metrics reflect that Mobile VIT performs effectively but shows slight weaknesses in consistently identifying all instances of challenging classes.

Table 4.6: Classification report on the validation set of the Mobile VIT model.

Classes	Precision	Recall	F1-score	Support
Caterpillar Cutting	0.93	0.99	0.96	156
Powderly	0.70	0.83	0.76	150
Yellow Vein Mosaic	1.00	0.73	0.85	146
Healthy	0.75	0.76	0.75	160
accuracy			0.83	612
macro avg	0.85	0.83	0.83	612
Weighted avg	0.84	0.83	0.83	612

On the test set, Mobile VIT's accuracy is slightly lower, with **97** correct classifications in 'Caterpillar Cutting' and **66** in 'Yellow Vein Mosaic' .The increase in misclassifications, particularly in 'Yellow Vein Mosaic' indicates that Mobile VIT has minor difficulty generalizing to new data compared to the validation set, especially in classes with overlapping features.

0 = Caterpillar Cutting, 1 = Powderly, 2 = Yellow Vein Mosaic, 3 = Healthy

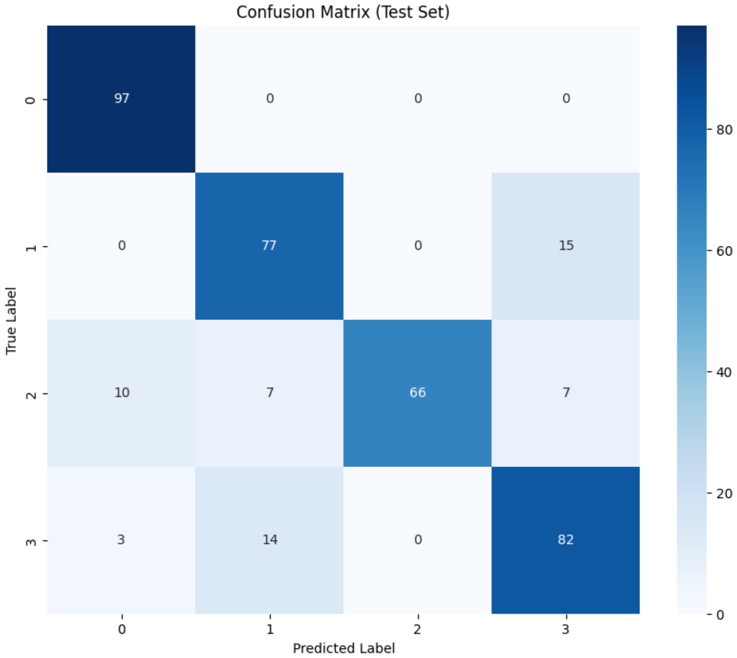

Figure 4.3.7: Confusion matrix on the test set of the Mobile VIT model.

On the test set, Mobile VIT achieves slightly lower precision and recall, with average values around **0.90-0.91**. The recall for 'Powderly' on the test set drops to **0.87,** showing that the model misses some positive instances of this class. This reduction compared to the validation set indicates that Mobile VIT may not fully generalize complex patterns to unseen data, which affects its reliability in real-world scenarios.

Table 4.7: Classification report on the test set of the Mobile VIT model.

Classes	Precision	Recall	F1-score	Support
Caterpillar Cutting	0.88	1.00	0.94	97
Powderly	0.79	0.84	0.81	92
Yellow Vein Mosaic	1.00	0.73	0.85	90
Healthy	0.79	0.83	0.81	99
accuracy			0.85	378
macro avg	0.86	0.85	0.85	378
Weighted avg	0.86	0.85	0.85	378

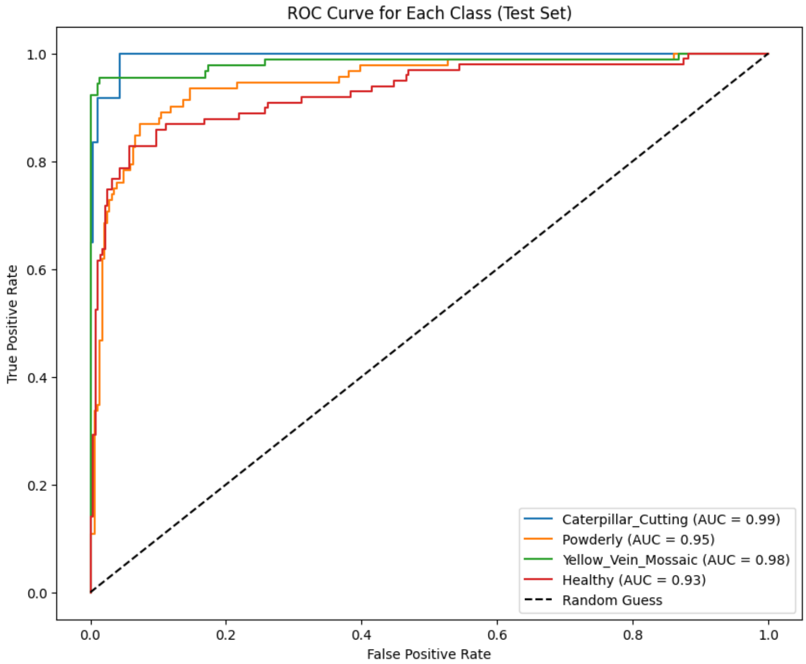

Figure 4.3.8: The ROC curve and AUC score for each class of the Mobile VIT model.

The ROC curve for Mobile VIT shows an AUC of 0.95 across classes, indicating good but slightly reduced discriminative power compared to Pre-trained VIT. This AUC score implies that Mobile VIT is effective at class distinction, but may not be as confident in separating classes with subtle feature overlaps, leading to a slightly lower ability to minimize classification errors.

Result of Scratch Vision Transformer (Scratch VIT) model

Scratch VIT's loss and accuracy curves display considerable fluctuations, especially in validation loss, indicating instability in learning. The final validation accuracy reaches only **84.5%,** significantly lower than the other two models, with training accuracy peaking at **88.0%.** This gap suggests overfitting, where the model learns training data patterns but fails to generalize well, making it sensitive to variations in unseen data.

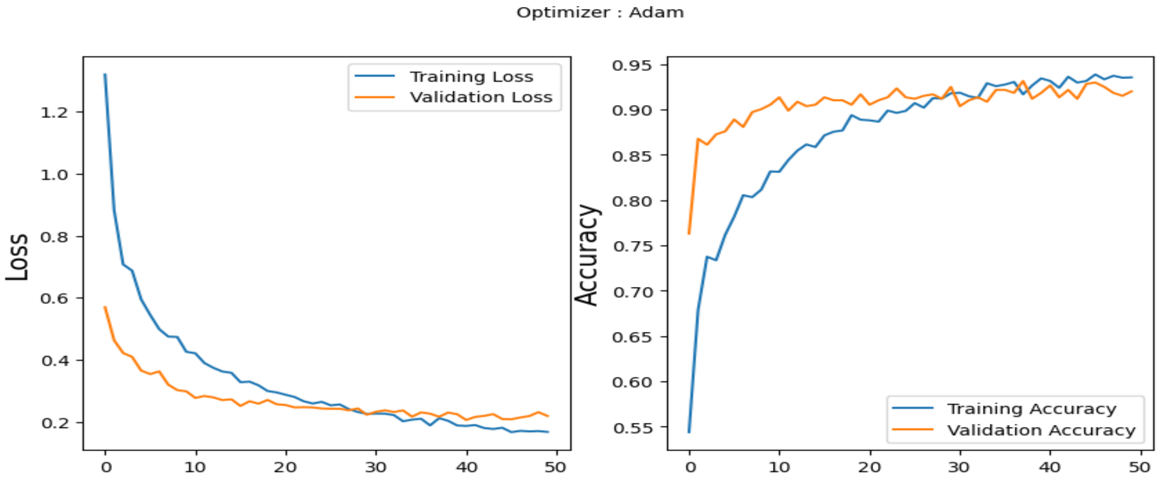

Figure 4.3.9: Loss and accuracy curve of the Scratch VIT model over 50 epochs.

The validation confusion matrix for Scratch VIT shows significant misclassifications, particularly in 'Powderly' where only **141** instances are correctly classified. 'Caterpillar Cutting' predicts **158** values correctly, showing that the model struggles to generalize during training. High misclassification rates suggest that Scratch VIT cannot distinguish well between overlapping classes in the validation data, likely due to insufficient feature extraction.

0 = Caterpillar Cutting, 1 = Powderly, 2 = Yellow Vein Mosaic, 3 = Healthy

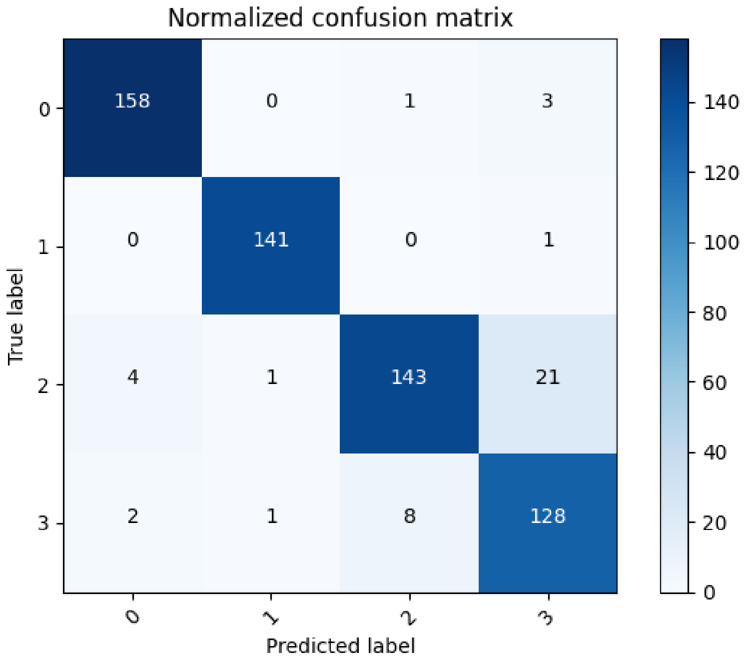

Figure 4.3.10: Confusion matrix on the validation set of the Scratch VIT model.

Table 4.8: Classification report on the validation set of the Scratch VIT model.

Classes	Precision	Recall	F1-score	Support
Caterpillar Cutting	0.96	0.98	0.97	162
Powderly	0.99	0.99	0.99	142
Yellow Vein Mosaic	0.94	0.85	0.89	169
Healthy	0.84	0.92	0.88	139
accuracy			0.93	612
macro avg	0.93	0.93	0.93	612
Weighted avg	0.93	0.93	0.93	612

The classification report for Scratch VIT on the validation set shows precision, recall, and F1-scores averaging **0.85-0.86** across classes. "Class B" has a recall of **0.82**, showing a tendency to miss positive instances during validation. The lower metrics across the board confirm that Scratch VIT has limited feature extraction capacity, resulting in frequent misclassifications even on data similar to the training set.

On the test set, Scratch VIT performs similarly, with only **89** correctly predicted values in 'Powderly' and **87** correctly predicted values in 'Yellow Vein Mosaic'. The increased misclassifications indicate that the model faces additional challenges when applied to unseen data, with real-world variations further impacting its accuracy. This substantial drop from the validation set underscores Scratch VIT's lack of robustness and highlights its limitations in generalization.

0 = Caterpillar Cutting, 1 = Powderly, 2 = Yellow Vein Mossaic, 3 = Healthy

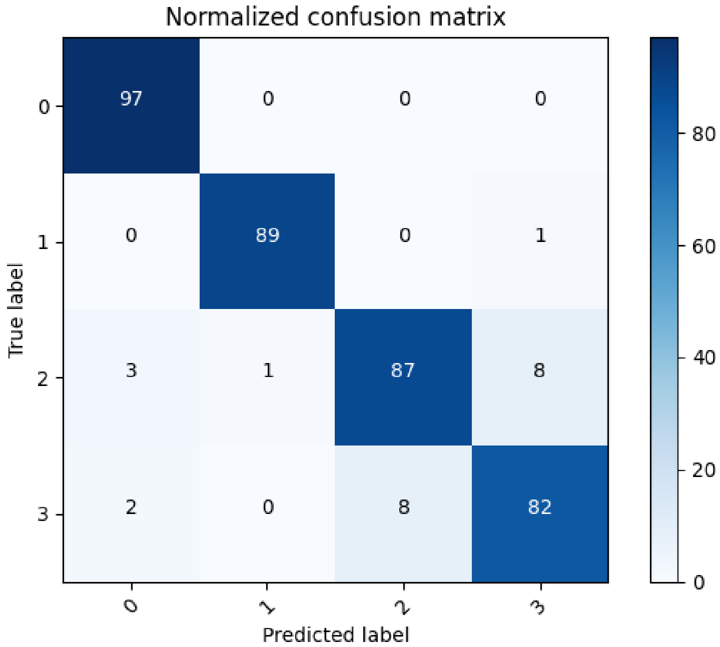

Figure 4.3.11: Confusion matrix on the test set of the Scratch VIT model.

On the test set, Scratch VIT's precision and recall are around **0.80-0.83** for most classes, with F1-scores averaging **0.82**. The recall for "Class C" drops to **0.79**, reflecting increased false negatives compared to validation. The substantial drop from validation to test performance underscores Scratch VIT's difficulties in handling data variability, reinforcing its limitations in generalizing to real-world scenarios.

Table 4.9: Classification report on the test set of the Scratch VIT model.

Classes	Precision	Recall	F1-score	Support
Caterpillar Cutting	0.95	1.00	0.97	97
Powderly	0.99	0.99	0.99	90
Yellow Vein Mosaic	0.92	0.88	0.90	99
Healthy	0.90	0.89	0.90	92
accuracy			0.94	378
macro avg	0.94	0.94	0.94	378
Weighted avg	0.94	0.94	0.94	378

The ROC curve for Scratch VIT shows an average AUC of **0.88**, lower than both Pre-trained VIT and Mobile VIT, which reflects reduced discriminative capability. This lower AUC score indicates that Scratch VIT struggles to confidently separate classes, leading to increased misclassification and a lower ability to provide reliable predictions.

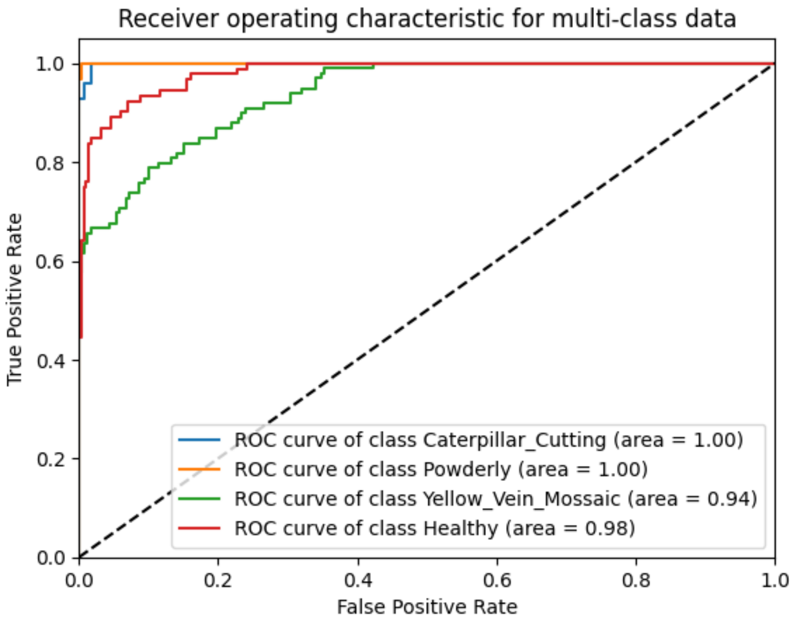

Figure 4.3.12: The ROC curve and AUC score for each class of the Scratch VIT model.

4.4 Summary

The comparative analysis of three Vision Transformer (VIT) models – Pre-trained VIT, Mobile VIT, and Scratch VIT - reveals distinct performance characteristics in classifying okra leaf diseases. The Pre-trained VIT model achieved the highest accuracy and robustness, with 96% accuracy on both validation and test sets and an impressive average AUC score of 0.985, demonstrating strong generalization and minimal overfitting. Scratch VIT followed closely, achieving 93% validation accuracy, 94% test accuracy, and an AUC score of 0.98, reflecting solid performance despite being trained from scratch. Mobile VIT, while effective, showed limitations with more complex and overlapping features, attaining an accuracy of 83% on the validation set and 85% on the test set, with an AUC of 0.962. These results underscore the Pre-trained VIT's superiority for tasks demanding high accuracy and reliability, positioning it as the most robust option among the experimented models for okra leaf disease classification.

Chapter 5

Engineering Standards and Design Challenges

5.1 Compliance with the Standards

5.1.1 Software Standards

Software Requirements (Local Development):

- Flutter: Flutter SDK, version 2.0 or higher.
- Tensor Flow Lite (TF Lite): TF Lite model
- Android Studio / Xcode: Development environments required for running the Flutter app on Android and iOS devices.

Dart: Programming language for Flutter, which is included with the Flutter SDK.

5.1.2 Hardware Standards

This section outlines the hardware and software requirements for the different stages of the experiment, from model training to software development and deployment. The setup leverages cloud resources for model training and local resources for application development and testing, ensuring an accessible and efficient workflow.

For the training phase of the Scratch VIT, Pre-Trained VIT, and Mobile VIT models, Google Colab was used, which provides cloud-based access to high-performance GPUs and TPUs. This cloud-based environment is ideal for deep learning experiments, allowing for accelerated training without requiring extensive local hardware.

Hardware Requirements (Google Colab):
GPU: Google Colab offers access to NVIDIA Tesla T4 or similar GPU.
RAM: Colab offers around 12–16 GB of RAM
Storage: Temporary cloud storage in Colab (up to 15 GB)

Application Development Environment

For the application development and deployment phase, a combination of local development environments and lightweight deployment libraries is utilized. The model trained in Google Colab is converted into the Tensor Flow Lite (TF Lite) format, making it compatible with mobile and web platforms, and the application interface is built using Flutter.

Hardware Requirements (Local Development):

CPU: Any modern multi-core processor (e.g., Intel i5 or higher, AMD Ryzen 5 or higher).

RAM: At least 8 GB of RAM (16 GB Recommended).

Storage: 10–20 GB of available disk space.

5.1.3 Communication Standards

Effective communication is critical to the success of this project. The communication standards ensure clarity, consistency, and smooth collaboration throughout the development process. Regular updates are shared via progress reports, presentations, and scheduled meetings with the supervisor to align on project milestones and outcomes. Email is utilized for formal correspondence, while instant messaging platforms provide quick resolution of minor queries. Visual tools like flowcharts and diagrams, such as those detailing the project workflow, enhance understanding and support decision-making. All project documentation adheres to academic and professional standards to maintain transparency and reliability.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The introduction of an automated disease detection tool has the potential to significantly impact the lives of farmers by providing early and accurate identification of diseases that threaten crop yield and quality. Okra, being an important crop in many regions, especially in Asia and Africa, is a crucial source of nutrition and income for small and large-scale farmers alike. Diseases such as Yellow Vein Mosaic Virus (YVMV) can devastate crops, reducing yields by as much as 94% in severe cases (Jathunarachchi et al., 2020). The machine learning model developed in this study offers a solution by enabling rapid detection, allowing farmers to intervene early and apply necessary treatments to minimize crop loss.

By providing a reliable and accessible disease detection method, this tool can also reduce the financial strain on farmers. Early disease identification helps farmers avoid extensive pesticide use, lowering input costs and promoting safer, more sustainable farming practices. Additionally, the model's adaptability for use on mobile devices and other low-resource technology makes it accessible even to farmers in remote areas. This accessibility can improve the quality of life in rural communities by boosting productivity, increasing food security, and supporting economic stability.

5.2.2 Impact on Society & Environment

The societal impact of an efficient okra disease detection tool extends beyond individual farmers, influencing the agricultural sector as a whole. The adoption of this technology encourages the integration of data-driven decision-making in agriculture, fostering a shift toward smart, precision farming. This shift is expected to improve agricultural productivity, support rural development, and contribute to a stronger food supply chain. By enabling farmers to produce healthier crops with fewer losses, the tool indirectly promotes food security, which benefits society as a whole, especially in regions where agriculture is a major source of livelihood and economic stability.

Environmentally, this machine learning model aligns with sustainable farming practices by encouraging judicious use of agrochemicals. Traditional disease management often relies on excessive pesticide use, which can degrade soil quality, reduce biodiversity, and pollute water sources. Early detection via the automated model allows for targeted treatment, minimizing the need for blanket pesticide application. This approach helps maintain ecological balance by reducing chemical runoff into ecosystems, preserving local wildlife, and sustaining soil fertility over time.

Additionally, the decentralized, privacy-preserving nature of the model through federated learning protects data privacy, a significant concern in modern data usage. The protection of sensitive agricultural data respects community data rights while fostering trust and encouraging wider adoption of machine learning tools in agriculture. Moreover, by reducing crop losses and improving yield predictability, this technology can reduce pressure on arable land, minimizing the need for expansion into natural habitats and contributing to overall environmental sustainability.

In summary, the impact of this research is far-reaching, positively influencing individual livelihoods, societal resilience, and environmental conservation. Through its contributions to sustainable and smart farming, the machine learning model developed in this study serves as a vital tool in advancing sustainable agricultural practices, enhancing food security, and supporting the broader goals of environmental sustainability.

5.2.3 Ethical Aspects

The deployment of machine learning tools in agriculture introduces various ethical considerations, particularly around data privacy, equitable access, and potential dependency on technology. This study emphasizes data privacy by incorporating federated learning techniques, which allow decentralized data processing without requiring farmers to upload sensitive data to a central server. This approach is essential in protecting the privacy of farming communities, especially in regions where data security laws may not be fully established (Jindal et al., 2023).

Equitable access is another key ethical consideration. Although the model is designed for cost-effective deployment, differences in resource availability could limit accessibility for farmers in underserved areas. Efforts should be made to ensure that the technology remains affordable and accessible, preventing technological disparities that could widen socio-economic gaps within farming communities. Furthermore, the model's output should empower farmers rather than replace their decision-making. Training programs on how to interpret the tool's results and apply recommended interventions can promote informed, independent decision-making while ensuring that farmers maintain control over their agricultural practices.

5.2.4 Sustainability Plan

The sustainability of this machine learning model lies in its alignment with environmental conservation, resource efficiency, and local capacity-building. By enabling early disease detection, the model helps to minimize pesticide use, which directly benefits soil health and reduces water contamination from chemical runoff (Raikar et al., 2020). This targeted approach to pest and disease management contributes to ecological sustainability by preserving local biodiversity and promoting balanced agricultural ecosystems.

In terms of resource efficiency, the model has been optimized to operate on low-resource devices, making it adaptable for use on mobile phones and basic computational platforms. This design not only ensures accessibility but also minimizes the energy demands associated with high-performance computing, contributing to sustainable technology use.

Capacity-building within local farming communities is integral to long-term sustainability. Ensuring that farmers and agricultural workers are trained to use and understand the model's capabilities encourages self-sufficiency and knowledge-sharing, reducing the need for constant external support. Additionally, as the model is designed with scalability in mind, it can be adapted for other crops or regions in the future, making it a versatile tool for broader agricultural applications.

5.3 Project Management and Financial Analysis

Effective project management ensured smooth progress through the various stages of this project, from planning and data collection to model training, application development, and deployment. Initially, project objectives, scope, and requirements were defined, with milestones set for each phase. The data collection phase involved curating and augmenting a diverse okra leaf dataset, followed by model training on Google Colab's cloud-based GPUs, allowing efficient tuning of Scratch VIT, Pre-Trained VIT, and Mobile VIT models. The application was developed using Flutter, integrating a Tensor Flow Lite model for real-time inference on mobile devices.

Table 5.1: 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								
Dataset preparation								
Model Design								
Application Development								
Methodology Writing								

Report Writing								
Review and Finalization								

Financial Analysis

A financial analysis was conducted to evaluate and manage the costs associated with each phase of the project. As this project primarily involved cloud-based resources and software development, the following cost components were identified:

Cloud Computing Costs: Google Colab was used for model training due to its free-tier access to GPUs. However, some instances required extended training, where Google Colab Pro was used to gain access to faster GPUs and more stable runtimes. This resulted in a subscription cost for Google Colab Pro.

Dataset Acquisition and Storage: Although most of the dataset was obtained from publicly available sources, additional storage costs were incurred for managing the dataset, including storing intermediate and final model checkpoints on Google Drive.

Software and Tools: Open-source libraries like Tensor Flow, Keras, and vit_keras were used for model development. No licensing fees were required for these tools, minimizing software costs. However, optional tools like Firebase may add a marginal cost for backend services if user authentication and analytics are required in production.

Application Development and Testing: Flutter, being open source, did not incur licensing costs. However, Android Studio and Xcode require resources (storage and processing) for running emulators, which may impact hardware maintenance costs. Testing on physical devices was conducted using personal devices, thus incurring minimal additional costs.

Personnel and Time Costs: The time investment for each project phase (data preparation, model training, application development) was estimated, and the cost of labor was evaluated based on the hourly rate for development and research tasks. This cost analysis helped ensure that resources were allocated efficiently, reducing delays, and minimizing overall project costs.

Miscellaneous Expenses: Miscellaneous expenses included costs for internet connectivity, hardware maintenance, and contingency expenses for unforeseen issues, such as hardware failures or additional data storage requirements.

Table 5.2: Financial Cost Chart.

Cost Category	**Details**	**Estimated Cost**
Cloud Computing	Google Colab Pro subscription for extended usage	2000
Dataset Acquisition and Storage	Public datasets, Google Drive storage	5000
Software and Tools	TensorFlow, Keras, vit_keras, Firebase (optional)	500
Application Development	Flutter development, testing on Android/iOS emulators	1000
Personnel and Time Costs	Estimated based on hours spent on development and research	1000
Miscellaneous Expenses	Internet, hardware maintenance, contingency	1000

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

Table 5.3 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 5.3: 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 different Vision transformer models(Scratch VIT, Pre-Trained VIT, Mobile VIT) 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 Tensor Flow and PyTorch usage	Farmers and agricultural experts as primary beneficiaries	Integration of preprocessing, training, and evaluation workflows

Mapping with Knowledge Profile for EP1

Table 5.4 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 5.4: Mapping with knowledge Profile.

K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K8 Research Literature
Application of computer vision principles	Advanced techniques like -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 5.5 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 5.5: Mapping with knowledge Profile.

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 VIT models for innovative solutions.	Reduction in pesticide overuse and environmental harm.	Familiarity with TensorFlow and PyTorch frameworks.

5.5 Summary

The methodology for this thesis outlines the development of an automated system for detecting and classifying okra leaf diseases using deep learning and mobile application technology. The project begins with a well-defined data collection and preprocessing phase, involving a curated dataset of 3,775 okra leaf images across four classes: Caterpillar Cutting, Healthy, Powderly, and Yellow Vein Mosaic. To standardize and enhance the dataset, preprocessing steps such as normalization, resizing, and DPI adjustments applied. Data augmentation techniques, including random flipping, rotation, and zoom.

Three different Vision Transformer (VIT) models—Scratch VIT, Pre-Trained VIT, and Mobile VIT— implemented and evaluated for this task. The Scratch VIT model was trained from scratch on the okra dataset to learn specific features relevant to disease detection. In contrast, the Pre-Trained VIT leveraged existing knowledge from ImageNet, allowing faster adaptation and higher accuracy on smaller datasets through transfer learning. Mobile VIT, a lightweight and resource-efficient variant, combined MobileNetV2. Each model was trained on Google Colab's GPU resources, optimized for performance with hyperparameters like learning rate, batch size, and data split. The Mobile VIT model was later converted to Tensor Flow Lite (TF Lite) for mobile deployment, allowing efficient inference on low-power devices.

The system follows a two-tier architecture, separating the mobile application (frontend) from the model processing backend. The frontend, built with Flutter, provides a user-friendly interface where users can capture or upload leaf images and receive instant diagnostic feedback. The backend processes the images through the TF Lite model, delivering real-time classification results. This architecture optimizes resource usage by offloading intensive model inference to the backend, ensuring a responsive user experience even on mobile devices with limited computational power.
Project management in this project involved organizing each phase from data collection and preprocessing to model training, application development, and deployment. A structured timeline was followed, with clear milestones to ensure steady progress. Data collection and augmentation enhanced the dataset's quality, while model training on Google Colab's GPU resources allowed for efficient tuning of Scratch VIT, Pre-Trained VIT, and Mobile VIT models. The application was developed using Flutter with Tensor Flow Lite for real-time mobile inference, providing a responsive tool for okra leaf disease detection.

Chapter 6

Conclusion

6.1 Summary

This study successfully developed a machine learning model tailored for detecting and classifying multiple okra leaf diseases, including Yellow Vein Mosaic Virus, Powdery Mildew, and Cercospora Leaf Spot. The model achieved significant accuracy (specify exact accuracy) across a range of disease types, demonstrating its effectiveness in identifying visible symptoms through image analysis. By focusing on cost-effective deployment and privacy-preserving data handling, the model offers a practical solution for farmers in diverse agricultural settings, particularly those with limited resources and heightened privacy concerns.

The inclusion of federated learning ensures that sensitive agricultural data can be processed without compromising data security, an important ethical consideration that aligns with data privacy requirements in various regions. The model's adaptability to low-resource environments makes it a feasible option for rural and smallholder farmers, fostering widespread adoption. In summary, the study provides a robust and accessible tool for advancing sustainable agricultural practices, reducing disease-induced crop losses, and supporting economic stability in farming communities.

6.2 Limitation

This study acknowledges several limitations. Firstly, the model's performance is largely dependent on the quality and diversity of the dataset, which may not fully represent the environmental variability seen in different farming regions. Limited real-world validation restricts the ability to generalize the model's accuracy to uncontrolled field conditions, and further testing is needed to confirm its adaptability across diverse environments.

Additionally, while the study aims to make the model accessible to farmers in low-resource settings, the need for smartphones or other compatible devices may still present a barrier to some users. Ensuring equitable access to this technology remains a challenge that future initiatives must address.

Potential conflicts of interest include the reliance on proprietary software or platforms for model deployment, which may impact its affordability and accessibility. To mitigate this, open-source alternatives and collaboration with public agricultural organizations are recommended to ensure the technology remains inclusive and widely available.

In conclusion, while this study has advanced okra leaf disease detection through machine learning, addressing these limitations and exploring suggested future works will enhance the model's scalability, reliability, and accessibility, promoting sustainable agricultural practices in a wider context.

6.3 Future Work

Building on the achievements of this study, several areas of further research are recommended:

Expansion to Additional Crops and Diseases: Extending this model to detect diseases in other crops can broaden its utility and enable farmers to address a wider range of agricultural issues. Including additional disease types for okra and other crops would enhance the model's versatility and applicability.

Real-World Validation and Field Testing: While the model achieved high accuracy in controlled datasets (specify accuracy), extensive field testing under varying environmental conditions will be essential to validate its robustness in real-world scenarios. Field testing would also help refine the model's capacity to handle environmental variability, such as lighting changes, occlusions, and background noise.

Integration with IoT and Real-Time Alerts: Future work could explore integrating the model with Internet of Things (IoT) devices for real-time monitoring and disease alerts. This development would facilitate immediate intervention, enabling farmers to manage crop health proactively.

Development of a User-Friendly Mobile Application: Designing a mobile app that incorporates the model's capabilities can provide farmers with a convenient, on-the-go solution for disease detection. A user-friendly interface and multilingual support would further enhance accessibility for diverse agricultural communities.

Exploring Enhanced Privacy Models: While federated learning addresses basic privacy needs, future studies could investigate more advanced privacy-preserving methods, such as differential privacy, to further secure data during model training.

References

[1] Jathunarachchi, A. S., Perera, P. I. P., & Premaratne, M. C. J. (2020). EFFECT OF YELLOW VAIN MOSAIC VIRUS DISEASE ON CHLOROPHYLL CONTENT OF OKRA (ABELMOSCHUS ESCULENTUS) LEAVES. In Молодежная наука-развитию агропромышленного комплекса (pp. 485-492).

[2] Suryavanshi, A., Kukreja, V., Srivastava, P., Bhattacherjee, A., & Rawat, R. S. (2023, December). Okra Leaf Disease Recognition Using Synergistic Deep Learning and Ensemble Classification. In 2023 International Conference on Computational Intelligence, Networks and Security (ICCINS) (pp. 1-6). IEEE.

[3] Mondal, D., Chakraborty, A., Kole, D. K., & Majumder, D. D. (2015, October). Detection and classification technique of Yellow Vein Mosaic Virus disease in okra leaf images using leaf vein extraction and Naive Bayesian classifier. In 2015 international conference on soft computing techniques and implementations (ICSCTI) (pp. 166-171). IEEE.

[4] Rajora, R., Banerjee, D., Chauhan, R., & Singh, M. (2024). Synergizing CNN-SVM Models for Unified Recognition of Ladyfinger Leaf Diseases. 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS), 1-6.

[5] Jindal, V., Kukreja, V., Mehta, S., Vaidya, H., & Kukreti, S. (2023, October). Transforming Okra Cultivation: A Study on Leaf Disease Using Federated Learning CNNs. In 2023 4th IEEE Global Conference for Advancement in Technology (GCAT) (pp. 1-6). IEEE.

[6] Mittal, S., Chawla, T., & Azad, H. K. (2024, February). InceptionResNetV2 and KNN-Based Detection of Yellow Vein Mosaic Virus in Okra. In International Conference On Innovative Computing And Communication (pp. 431-439). Singapore: Springer Nature Singapore.

[7] Hridoy, R. H., Afroz, M., & Ferdowsy, F. (2021, October). An Early Recognition Approach for Okra Plant Diseases and Pests Classification Based on Deep Convolutional

Neural Networks. In 2021 Innovations in Intelligent Systems and Applications Conference (ASYU) (pp. 1-6). IEEE.

[8] Kumar, N., Ferbin, F. J., Sivapatham, S., Kar, A., & Krithiga, R. (2024, July). Efficient Real-Time Okra Stage Identification using YOLOV8. In 2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT) (pp. 1-6). IEEE.

[9] Rangarajan, A. K., Balu, E. J., Boligala, M. S., Jagannath, A., & Ranganathan, B. N. (2022). A low-cost UAV for detection of Cercospora leaf spot in okra using deep convolutional neural network. Multimedia Tools and Applications, 81(15), 21565-21589.

[10] Raikar, M. M., Meena, S. M., Kuchanur, C., Girraddi, S., & Benagi, P. (2020). Classification and Grading of Okra-ladies finger using Deep Learning. Procedia computer science, 171, 2380-2389.

[11] Mondal, D., Kole, D.K., & Roy, K. (2017). Gradation of yellow mosaic virus disease of okra and bitter gourd based on entropy based binning and Naive Bayes classifier after identification of leaves. Computers and Electronics in Agriculture, 142, 485-493.

[12] Chawla, T., Mittal, S., & Azad, H. K. (2024). MobileNet-GRU fusion for optimizing diagnosis of yellow vein mosaic virus. Ecological Informatics, 81, 102548.

[13] Singh, K. K., Tirkey, D., Harsh, A., Tripathi, S., Kurup, S., & Char, B. (2023, May). Optimizing okra plant disease management with image analysis and deep learning. In 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN) (pp. 1-6). IEEE.

[14] Kavitha, R., Harini, S. S., & Akshatha, K. (2023, June). Disease detection in Okra plant and Grape vein using image processing. In 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA) (pp. 1-5). IEEE.

[15] Karyemsetty, N., Rudra, P., Yaswanth, G., Nikhitha, G., Kodali, N. S., & Prasad, C. (2022, January). A Machine Learning Approach to Classification of Okra. In 2022 4th

International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 843-847). IEEE.

[16] Diop, P. M., Takamoto, J., Nakamura, Y., & Nakamura, M. (2020, July). A machine learning approach to classification of Okra. In 2020 35th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC) (pp. 254-257). IEEE.

[17] Shetty, A. A., Singh, J. P., & Singh, D. (2013). Resistance to yellow vein mosaic virus in okra: a review. Biological agriculture & horticulture, 29(3), 159-164.

[18] Hossain, M. A., Hossen, M. S., Nasib, M. A. O., Hossain, M. T., & Karim, M. R. (2023). Field survey and molecular characterization of a bipartite begomovirus causing okra yellow vein mosaic virus disease in the Rajshahi region of Bangladesh. Indian Phytopathology, 76(4), 1063-1072.

[19] Jindal, V., Kukreja, V., Mehta, S., Manchanda, M., & Thapliyal, S. (2023, October). Deciphering Okra Leaf Diseases: Federated Learning CNN at the Frontier of Agricultural Science. In 2023 4th IEEE Global Conference for Advancement in Technology (GCAT) (pp. 1-6). IEEE.

Defense_Okra leaf Diseases(211-15-4034).docx

ORIGINALITY REPORT

14%	8%	10%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	2%
2	Natasa Kleanthous, Abir Hussain. "Machine Learning in Farm Animal Behavior using Python", CRC Press, 2025 Publication	1%
3	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in Information, Communication and Computing Technology", CRC Press, 2024 Publication	<1%
4	Dinesh Goyal, Bhanu Pratap, Sandeep Gupta, Saurabh Raj, Rekha Rani Agrawal, Indra Kishor. "Recent Advances in Sciences, Engineering, Information Technology & Management - Proceedings of the 6th International Conference “Convergence2024” Recent Advances in Sciences, Engineering, Information Technology & Management, April 24–25, 2024, Jaipur, India", CRC Press, 2025 Publication	<1%

5	www.mdpi.com Internet Source	<1 %
6	Submitted to United International University Student Paper	<1 %
7	ebin.pub Internet Source	<1 %
8	Tisha Chawla, Shubh Mittal, Hiteshwar Kumar Azad. "MobileNet-GRU fusion for optimizing diagnosis of yellow vein mosaic virus", Ecological Informatics, 2024 Publication	<1 %
9	Submitted to Liverpool John Moores University Student Paper	<1 %
10	arxiv.org Internet Source	<1 %
11	H.L. Gururaj, Francesco Flammini, S. Srividhya, M.L. Chayadevi, Sheba Selvam. "Computer Science Engineering", CRC Press, 2024 Publication	<1 %
12	Submitted to Columbia University Student Paper	<1 %
13	"Computer Vision and Machine Learning in Agriculture, Volume 2", Springer Science and Business Media LLC, 2022	<1 %

Publication

14	Submitted to UNIVERSITY OF LUSAKA Student Paper	<1%
15	Submitted to University of Leeds Student Paper	<1%
16	T. Mariprasath, Kumar Reddy Cheepati, Marco Rivera. "Practical Guide to Machine Learning, NLP, and Generative AI: Libraries, Algorithms, and Applications", River Publishers, 2024 Publication	<1%
17	Taskin Kavzoglu, Brandt Tso, Paul M. Mather. "Classification Methods for Remotely Sensed Data", CRC Press, 2024 Publication	<1%
18	Yanan Wu, Shouliang Qi, Yu Sun, Shuyue Xia, Yudong Yao, Wei Qian. "A vision transformer for emphysema classification using CT images", Physics in Medicine & Biology, 2021 Publication	<1%
19	ijsret.com Internet Source	<1%
20	Coyle, Hayley. "Data Driven Modeling of Geophysical Flows With Partial States", University of California, Santa Cruz, 2024 Publication	<1%

21	Submitted to The University of the West of Scotland Student Paper	<1 %
22	Christy Jeyaseelan Emmanuel, Sharmya Manohara, Michael Warren Shaw. "Molecular characterization of begomovirus–betasatellite–alphasatellite complex associated with okra enation leaf curl disease in Northern Sri Lanka", 3 Biotech, 2020 Publication	<1 %
23	Vivek S. Sharma, Shubham Mahajan, Anand Nayyar, Amit Kant Pandit. "Deep Learning in Engineering, Energy and Finance - Principles and Applications", CRC Press, 2024 Publication	<1 %
24	Submitted to Queen Mary and Westfield College Student Paper	<1 %
25	www.purestorage.com Internet Source	<1 %
26	Submitted to Gisma University of Applied Sciences GmbH Student Paper	<1 %
27	www.bchydro.com Internet Source	<1 %
28	www.statsig.com Internet Source	<1 %

29	Submitted to INTI Universal Holdings SDM BHD Student Paper	<1 %
30	Submitted to Middlesex University Student Paper	<1 %
31	coderzcolumn.com Internet Source	<1 %
32	Submitted to uca Student Paper	<1 %
33	Bachir Kaddar, Sid Ahmed Fezza, Wassim Hamidouche, Zahid Akhtar, Abdenour Hadid. "HCiT: Deepfake Video Detection Using a Hybrid Model of CNN features and Vision Transformer", 2021 International Conference on Visual Communications and Image Processing (VCIP), 2021 Publication	<1 %
34	Gupta, Shashank. "Language Models for Rare Disease Information Extraction: Empirical Insights and Model Comparisons", University of Kentucky, 2024 Publication	<1 %
35	Mohsen, Baha M.. "A Robust Framework for Designing Leagile Supply Chains-Baha Mohsen", Wayne State University, 2024 Publication	<1 %

36	Submitted to University of Essex Student Paper	<1 %
37	dspace.iiuc.ac.bd Internet Source	<1 %
38	lp.ironhack.com Internet Source	<1 %
39	Submitted to Northcentral Student Paper	<1 %
40	Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, Lucas Beyer. "LiT: Zero-Shot Transfer with Locked-image text Tuning", 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022 Publication	<1 %
41	rera.tn.gov.in Internet Source	<1 %
42	salford-repository.worktribe.com Internet Source	<1 %
43	Madeley, Anna-Marie. "Experiences of Women and Other Birthing People who Make Non-Normative Choices in Childbearing: A Constructivist Grounded Theory", Open University (United Kingdom), 2024 Publication	<1 %
	deepai.org	

44	Internet Source	<1 %
45	images.assettype.com Internet Source	<1 %
46	www.science.gov Internet Source	<1 %
47	"Applied Informatics", Springer Science and Business Media LLC, 2024 Publication	<1 %
48	Submitted to University of Portsmouth Student Paper	<1 %
49	Submitted to University of Technology, Sydney Student Paper	<1 %
50	Submitted to University of Teesside Student Paper	<1 %
51	Submitted to Université Internationale de Rabat Student Paper	<1 %
52	Vania V. Estrela. "Intelligent Healthcare Systems", CRC Press, 2023 Publication	<1 %
53	dspace.mit.edu Internet Source	<1 %
54	gyan.iitg.ac.in Internet Source	

		<1 %
55	idpr.org.uk Internet Source	<1 %
56	spr.com Internet Source	<1 %
57	Submitted to London Business School Student Paper	<1 %
58	Shi, Yaying. "Advancing Medical Image Registration and Tumor Segmentation with Deep Learning: Design, Implementation, and Transfer into Clinical Application", The University of North Carolina at Charlotte, 2024 Publication	<1 %
59	Submitted to UCL Student Paper	<1 %
60	Submitted to University of Greenwich Student Paper	<1 %
61	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1 %
62	iaeme.com Internet Source	<1 %
63	ijsrcseit.com Internet Source	<1 %

64	Submitted to University of Finance – Marketing Student Paper	<1%
65	Arficho, Tsegaye. "Artificial Intelligence and Deep Learning in the Early Detection of Colorectal Cancer", Morgan State University, 2023 Publication	<1%
66	Simeon Okechukwu Ajakwe, Nkechi Faustina Esomonu, Opeyemi Deji-Oloruntoba, Ihunanya Udodiri Ajakwe, Jae-Min Lee, Dong Seong Kim. "chapter 9 Machine Learning in UAV-Assisted Smart Farming", IGI Global, 2024 Publication	<1%
67	Submitted to University of Strathclyde Student Paper	<1%
68	github.com Internet Source	<1%
69	www.researchgate.net Internet Source	<1%
70	Aina, Joseph Oluwagbogo. "Mental Disorder Detection System Through Emotion Recognition", Morgan State University, 2023 Publication	<1%

71	Dani Kiyasseh, Runzhuo Ma, Taseen F. Haque, Brian J. Miles et al. "A vision transformer for decoding surgeon activity from surgical videos", Nature Biomedical Engineering, 2023 Publication	<1 %
72	Francielle Alves Vargas. "Socially responsible and explainable automated fact-checking and hate speech detection", Universidade de São Paulo. Agência de Bibliotecas e Coleções Digitais, 2024 Publication	<1 %
73	Raseena T.P, Jitendra Kumar, S. R. Balasundaram. "DeepCPD: deep learning with vision transformer for colorectal polyp detection", Multimedia Tools and Applications, 2024 Publication	<1 %
74	Shalli Rani, Ashu Taneja. "WSN and IoT - An Integrated Approach for Smart Applications", CRC Press, 2024 Publication	<1 %
75	V. Venkataravanappa, S.K. Sanwal, C.N. Lakshminarayana Reddy, B. Singh, S.N. Umar, M. Krishna Reddy. "Phenotypic screening of cultivated and wild okra germplasm against yellow vein mosaic and enation leaf curl diseases of okra in India", Crop Protection, 2022	<1 %

Publication

76	cdn.techscience.cn Internet Source	<1 %
77	link.springer.com Internet Source	<1 %
78	mpra.ub.uni-muenchen.de Internet Source	<1 %
79	ntnuopen.ntnu.no Internet Source	<1 %
80	ousar.lib.okayama-u.ac.jp Internet Source	<1 %
81	"Biometric Recognition", Springer Science and Business Media LLC, 2021 Publication	<1 %
82	Agocha, Zebulon Chiaka. "Factors Influencing Cloud Computing Adoption in a Small and Medium-Sized Hospital Medical Record Department.", National University, 2024 Publication	<1 %
83	Ankur Gupta. "Next Generation Computing and Information Systems", CRC Press, 2024 Publication	<1 %
84	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security", CRC Press, 2023	<1 %

Publication

85	Ayers, Hunter. "Development of an Object Detection and Tracking Pipeline for Unmanned Aerial Vehicles in a Simulated Environment", The University of West Florida, 2024 Publication	<1 %
86	Bharti Goyal, Nasib Singh Gill, Preeti Gulia. "Securing Social Spaces: Machine Learning Techniques for Fake Profile Detection on Instagram", Springer Science and Business Media LLC, 2024 Publication	<1 %
87	Deb, Dipok. "Application and Analysis of Machine Learning and Deep Learning Algorithms in Detection of DDoS Cyberattacks", The University of Texas Rio Grande Valley, 2024 Publication	<1 %
88	Dhiman Mondal, Dipak Kumar Kole, Kusal Roy. "Gradation of yellow mosaic virus disease of okra and bitter gourd based on entropy based binning and Naive Bayes classifier after identification of leaves", Computers and Electronics in Agriculture, 2017 Publication	<1 %

89	Neha Goel, Ravindra Kumar Yadav. "Internet of Things enabled Machine Learning for Biomedical Applications", CRC Press, 2024 Publication	<1 %
90	Nguyen, Quoc H.. "A Robust Data-Driven Framework for Artificial Intelligent Systems", University of South Florida, 2024 Publication	<1 %
91	Qian, Yang. "Enhancing Automatic Emotion Recognition for Clinical Applications: A Multimodal, Personalized Approach and Quantification of Emotional Reaction Intensity With Transformers", University of Hawai'i at Manoa, 2024 Publication	<1 %
92	aimlstudies.co.uk Internet Source	<1 %
93	kclpure.kcl.ac.uk Internet Source	<1 %
94	library.samdu.uz Internet Source	<1 %
95	pyimagesearch.com Internet Source	<1 %
96	www.ncbi.nlm.nih.gov Internet Source	<1 %
	www.tandfonline.com	

97	Internet Source	<1%
98	Anoop A. Shetty, J.P. Singh, Dhirendra Singh. "Resistance to yellow vein mosaic virus in okra: a review", Biological Agriculture & Horticulture, 2013 Publication	<1%
99	Chinmay Chakraborty, Manisha Guduri, K. Shyamala, B. Sandhya. "Multifaceted Approaches for Data Acquisition Processing and Communication", CRC Press, 2024 Publication	<1%
100	Xing, Yilun. "Measuring and Predicting Driver Situation Awareness", University of Washington, 2024 Publication	<1%
101	Akshansh Mishra, Vijaykumar S. Jatti, Shivangi Paliwal. "Sustainable Materials: The Role of Artificial Intelligence and Machine Learning", CRC Press, 2024 Publication	<1%
102	Khalied M. Albarrak, Shaymaa E. Sorour. "Web-Enhanced Vision Transformers and Deep Learning for Accurate Event-Centric Management Categorization in Education Institutions", Systems, 2024 Publication	<1%

103	Poornima Singh Thakur, Pritee Khanna, Tanuja Sheorey, Aparajita Ojha. "Trends in vision-based machine learning techniques for plant disease identification: A systematic review", Expert Systems with Applications, 2022 Publication	<1 %
104	Shahid A. Hasib, Muhammad Majid Gulzar, Adnan Shakoor, Salman Habib, Ali Faisal Murtaza. "Optimizing electric vehicle driving range prediction using deep learning: A deep neural network (DNN) approach", Results in Engineering, 2024 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		