AgriViTX: An Explainable Vision Transformer Model for Multi-Crop Disease Detection with Farmer-Centric Deployment

By
Rokonozzaman Ayon
213-15-4393

Abdullah Al Rahat
213-15-4261

FINAL YEAR DESIGN PROJECT REPORT

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

Supervised by

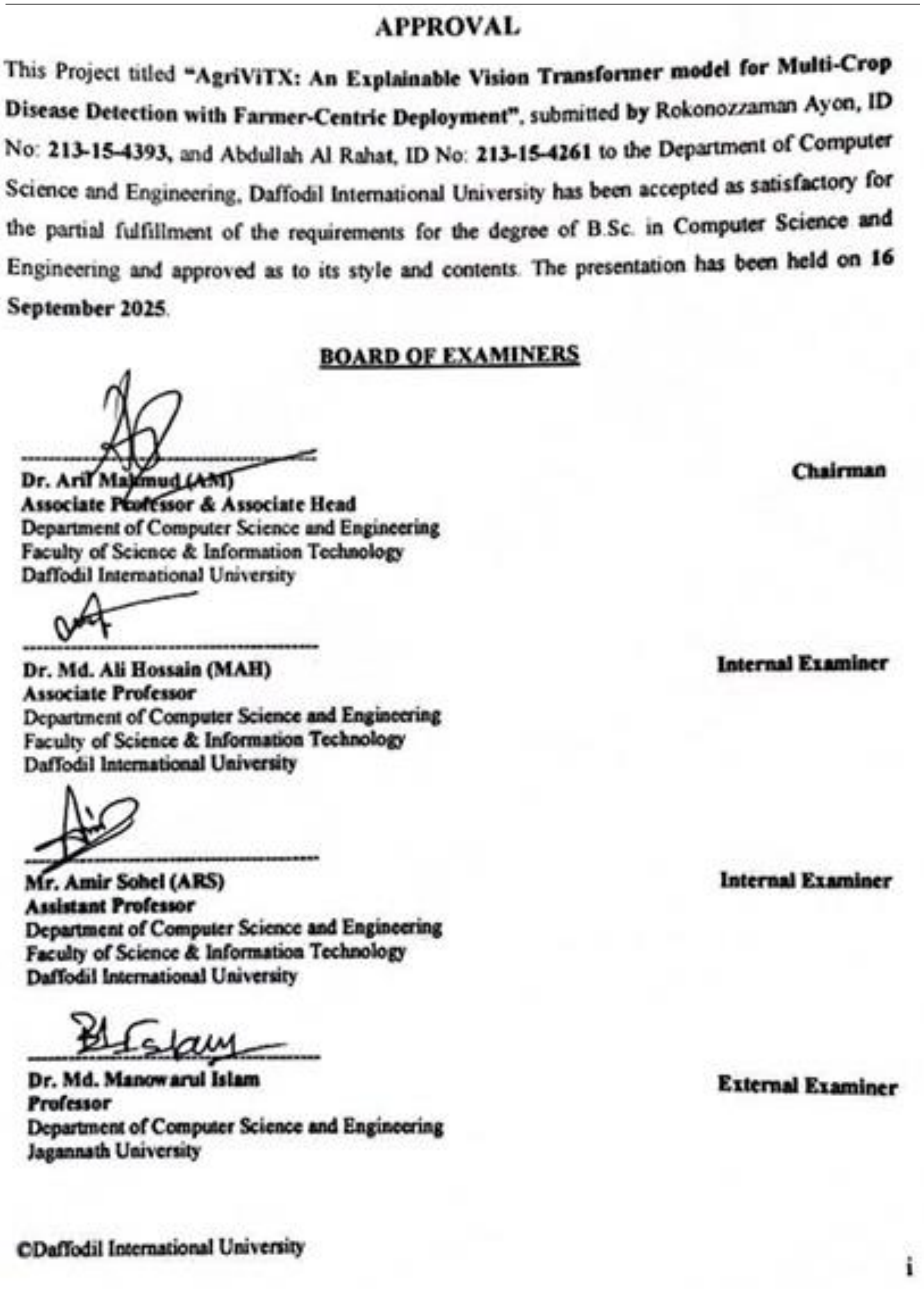
Dr. Md. Monzur Morshed
Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

Sharun Akter Khushbu
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

September 16, 2025



APPROVAL

This Project titled **"AgriViTX: An Explainable Vision Transformer model for Multi-Crop Disease Detection with Farmer-Centric Deployment"**, submitted by Rokonozzaman Ayon, ID No: **213-15-4393**, and Abdullah Al Rahat, ID No: **213-15-4261** 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 **16 September 2025**.

BOARD OF EXAMINERS

Dr. Arif Mahmud (AM)
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman

Dr. Md. Ali Hossain (MAH)
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Mr. Amir Sohel (ARS)
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Md. Manowarul Islam
Professor
Department of Computer Science and Engineering
Jagannath University

External Examiner

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. Md. Monzur Morshed**, **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:

Dr. Md. Monzur Morshed
Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:
7.9.25

Sharun Akter Khushbu
~~Senior Lecturer~~ Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Submitted by:

Rokonozzaman Ayon
Student ID: 213-15-4393
Department of Computer Science and Engineering
Daffodil International University

Rahat

Abdullah Al Rahat
Student ID: 213-15-4261
Department of Computer Science and Engineering
Daffodil International University

ACKNOWLEDGEMENTS

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

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

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

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

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

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

ABSTRACT

Crop diseases cause great risks to world food security and livelihood of farmers particularly in the developing nations such as Bangladesh where maize, tomato and onion are staple crops. The conventional manual detection systems are inaccessible and slow with many errors and this highlights the importance of scalable AI-based solutions. This work suggests a deep learning model to perform multi-crop disease detection with Convolutional Neural Networks (CNNs) and Vision Transformer (ViTs). A number of baseline CNN models were compared to a custom ViT architecture where their performance was measured in accuracy, precision, recall, and F1-score. The experiments showed that the ViT was better than CNN baselines with an individual dataset accuracy of 98% on tomato, 96% on onion, and 97% on maize. In the case of the multi-crop classification carried out when the datasets were pooled, the ViT model achieved a higher overall accuracy of 98.7% which shows good generalization across crops. To better interpretability the use of pseudo-segmentation methods was undertaken, where the specific disease-affected areas which are highlighted by the model could be visualized. In addition, an operational web application was created to allow identifying diseases in time when a user uploaded a leaf image, which should be provided as a useful tool to farmers and agricultural advisors. In the course of evaluation, Explainable AI (XAI) tools like LIME, and SHAP were implemented, but their integration into deployed systems is still a matter of future work. Altogether, the study confirms the usefulness of Vision Transformers in terms of strong, explainable, and convenient detection of crop diseases and offers a base in the development of mobile devices that could be used in the future to help farmers to sustain their agriculture.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The initial section of this chapter provides a background, asserting why the detection of crop diseases is important, and why it is particularly important in Bangladesh and South Asia. It then specializes in three crops, maize, tomato, and onion, and explains why each of these is so significant in local diets and economies. Then, the author goes through the key challenges that may arise when disease outbreaks occur, mentions some of the past attempts to address these issues, and outlines why actions based on deep learning are receiving newfound interest. Then, the section swings into two emerging models: Convolutional Neural Networks (CNN) and Vision Transformers (ViT). The chapter proceeds to discuss the functioning of both models, runs through their advantages and drawbacks, and presents a comparison of the two. Lastly, the chapter concludes by further elaborating on how early detection of diseases in farming activities could enhance agricultural productivity should these models be utilized effectively.

1.1 Introduction

Agriculture in most cases in developing economies particularly in South Asia forms the mainstay of such economies as it sustains huge populations through this form of employment. To take an example, in Bangladesh, over 40 percent of the laborers in the country depend on agricultural activities, and the sector holds an approximate value of 12.7 percent in the overall percentage of national Gross Domestic Product in the year 2023 [1,2]. More important crops, which include maize, tomato, and onion are dominant since they are actively economically rewarding and of high culture. However, due to the high acreage in which such crops are actual grown, the crops have been subjected more and more to fungal, bacterial and viral diseases which erode their yearly output. Studies indicate that crop diseases have the capacity to reduce yields of South Asian agriculture by up to 2030 percent per year and some outbreaks can result in even higher damages [3].

In Bangladesh, maize (corn) production nationally is now coming a close second only behind rice as a cereal crop [4]. It is vital in the manufacture of animal feeds and is finding its way into many food-processing activities. Maize production has increased at an impressive compound annual growth rate (CAGR) of around 12% based on statistics that were gathered by the Bangladesh Bureau of Statistics (BBS) with maize production standing at approximately 1.2 million metric tons in 2010 and over 5 million metric tons in 2021 [5]. Although this performance is noteworthy, maize farming is susceptible to a complex of diseases, which include Northern leaf blight, Gray leaf spot, and Maize lethal Necrosis in particular [6]. Whenever such pathogens attack, crop production can be reduced drastically and this fact is borne upon small hold farmers, typically putting their livelihoods and the local agricultural systems.

Tomato is a first-rate winter vegetable to many of us living in central and southern Bangladesh. It has helped to feed our families and helps to boost our nations supply of vegetables. In tandem, last year alone Bangladesh exported 482,000 metric tons of eventual setting tomato to make it the sixth-largest vegetable export, constituting more than 2 percent of the overall farmers income [7]. Sadly, tomato farms are devastated by Early Blight, Late Blight, Bacterial Wilt, and Leaf Curl Virus which not only reduces the attractiveness and production, but also reduces the market worth [8]. As the basic economy of many rural communities involves small- and medium-scale farming, an outbreak of any of these diseases will result in severe economic losses.

The onion is simultaneously the most popular and yet one of the most uncommonly grown produce in Bangladesh and yet locally the onion is primarily grown in the northwestern sides of the country with the demands taking precedence and therefore leaving the local produce with little to chew on other than to import onions in the other countries [9]. Domestic production of onion by farmers in 2023 alone has been estimated at about 2.5-million metric tons, and the nation was conducting imports of around USD 200-million worth of the same products to lag behind [10]. A triple threat to Purple Blotch, Downy Mildew and Basal Rot, it can destroy yields, and reset the clock regarding the storage life, is among the greatest contributors to said deficiencies [11]. Add a couple of current shaky supplies chains and the bucketing and crashing of global prices and all of a sudden farmers are feeling the heat, to get smarter at diagnosing disease in early stages because that is the only way they can get to curtail post-harvest, on the one hand and keep the market on the other from unnerving itself again [12].

Early detection of plant diseases has been identified as an important factor of maintaining stable agricultural production particularly in places where agriculture contributes significantly to the food security system, and income generating activity [13]. Once crop diseases go unnoticed or uncontrolled in their incipient phases, one might notice massive loss of yield and in some reports, the losses might range up to 30 percent in the regions that have been hit [14]. Such losses do not only decrease the supply of much needed food crops, but also affect the financial security of the farmers-who in many cases work at a subsistence level. Furthermore, failure to diagnose indiscrimination of diseases also leads to the overuse or wrong application of chemical pesticides that in addition to raising production costs puts a strain on the immediate environment and on the wellbeing of the consumers [15]. However, in conventional agriculture, the identification of diseases has been largely based on visual means performed by agricultural specialists or a skilled farmer [16]. This manual method is however prone to errors by humans and tends to be uneven in diverse scales of production.

Convolutional Neural Networks (CNNs) have become one of the most popular architectures of image-based disease classification among other deep learning methods. InceptionV3, ResNet50, Xception, and InceptionResNetV2 CNN systems have been applied in some agricultural research studies to do plant disease classification based on the symptom appearance on the plants leaf [17 18]. These models can identify major characteristics of the disease such as lesions or color changes, and deformations and surface textures. They work based on extracting local characteristics of various components of an image based on a hierarchy of convolution. This has led to the popularity of CNNs in the task of crop disease detection because of their accuracy and efficiency when classifying localized patterns [18]. Nevertheless, there are still some limitations inherent in CNNs which are capable of having an impact on their performance, especially on any complex or multi-crop data. Among the key limitations is the fact that they do not represent the long-range dependence and global spatial information in image very well [19]. This is as a result of the limited receptive field in each convolutional layer thus CNNs only take into consideration small local patches at a time [20].

With the above shortcomings of CNNs, a promising method named the Vision Transformer (ViT) has been proposed as an alternative to apply to image-related classification activities. The transformer architecture has been designed initially to handle natural language processing and can also be used in computer vision. In contrast to CNNs where the spatial features can be extracted using the convolutional operation, the ViTs process the input image in the form of fixed-size image patches as part of the sequence. Each patch is informed to a vector and the model is trained to discover connections between every patch with the help of an auto-attention mechanism. This makes the model able to capture global dependencies in the image no matter the physical distance between features.

ViTs capabilities of learning long-range contextual data specifically support multi-class disease identification in crops where the manifestation of the disease is often not consistent or localized. Such relationships can be more successfully identified with the use of ViT model compared to conventional CNNs. The multi-crop disease classification problem has been proposed with a solution of using ViT-based deep learning in a research study. The model was used across three datasets on maize, tomato and onion, which had samples of different types of disease as well as healthy ones. These datasets were taken through an open depository and exhibited enough heterogeneity to test out the integrity and the generalization capability of the proposed model. In its attempt to prove its efficiency, the ViT architecture was compared with six popularly established CNN-based models including InceptionV3, DenseNet201, InceptionResNetV2, ResNet101, ResNet152V2 and Xception. The comparative study was conducted to evaluate the performance of each of the models on the crops of various types and determine whether a single ViT-based architecture may be successfully applied to generalize all three datasets.

The final goal of this study is to come up with a fast and efficient scalable model that is easy to implement in a practical agricultural environment, especially in resource limited areas such as the one in rural Bangladesh. The proposed system may help to overcome all these factors by facilitating detection of diseases in maize, tomato, and onion early and in a precise manner to minimize yield loss, unnecessary use of chemical treatment, and enhance totality of crop management measures. ViT applied here can not only be a new perspective in applying transformer models in agriculture with all the support it gives to other research areas but also to pursue new directions in research using uniform systems in multi-crop disease classification in the future.

1.2 Motivation

One of the economic and socio-economic aspects of the food security and livelihoods in Bangladesh is crop production. Maize, tomato, and onion, being high yielding and becoming economically significant to the country, are one of the crops that are highly developed with small and medium-scale farmers widely producing them in various regions of the country.

Bangladesh Bureau of Statistics (BBS) recorded that over the recent years, the maize production has been over 5 million metric tons due to the increasing demand in the poultry industry and the food processing industry. In the same way, tomatoes have exceeded 480,000 metric tons production, with the output being used by both the domestic and the export market [21]. Onion, though grown on an area of more than 240,000 hectares, relies heavily on imports- a testament to its popularity as well as susceptibility. Taken together, these crops help to make millions of farming households in Bangladesh and South Asia economically secure [22]. The increased rate of fungal, bacterial and viral infections in these crops has however become a big challenge. Research indicates that crop diseases may cut yield by 20 to 30 percent per year, threatening food security and revenues of farmers in case they are not detected and suppressed on time [23].

Due to the inaccessibility of agricultural advisory services in most of the rural regions, automated and dependable systems of plant disease detection are now vital. Recognizing crop diseases early may lead to the prevention of severe outbreaks, minimize the excessive use of pesticides, and enhance more accurate actions that would positively impact the environment and economy. The urgency of the problem that led to this research was to provide farmers with a technological solution that can be implemented even in a poor resource setting.

This paper will discuss how to tackle these limitations by examining the recent and potent architecture called Vision Transformers (ViT), which is based on the transformer framework applied to computer vision problems. In contrast to CNNs, the ViT models can treat the whole image as a list of patches and learn long-range spatial connections, which is appropriate to classify diseases in multiple crops with different symptoms distributions. In the present study, the authors have suggested a lightweight ViT-based model that will be efficient and effective in all three datasets (maize, tomato, and onion).

1.3 Objectives

The main objective of the study is to create a lightweight, efficient and generalizable deep learning-based disease detection system in three of the most economically crucial crops in Bangladesh, namely, maize, tomato and onion. The aim is to show how the suggested solution, as the model employing the Vision Transformer (ViT) architecture, will be capable of a high level of accuracy on a variety of crop datasets, whereas it can still be applicable in a limited-resource rural environment. The work also intends to create a basis of integrating interpretable AI processes in the future where one can make trustworthy and explainable decisions within the agricultural disease diagnosis sphere.

The following detailed objectives guide the full scope of this research:

- To build and construct an efficient, transformer structure with the vision transformer (ViT) model, able to identify several diseases in a variety of crops. This model is chosen because it can ensure such global context and long-range dependencies in plant leaf images, as well as alleviate the weaknesses of the conventional CNNs in practical agricultural use.

- To build the foundation of future integration of the Explainable AI (XAI) methods like Pseudo Segmentation, LIME, and SHAP into the prediction pipeline of the ViT model.

- To build a web-based application using the proposed AgriViTX model, where AI is combined with explainable methods to give clear and user-friendly results. The system will help farmers and researchers by providing accurate disease predictions with visual explanations, along with suggested actions farmers should take based on the detected disease. In addition, users will be able to download a detailed report in PDF format for future reference and support.

1.4 Methodology

The research process of the current study is based on a multi-step methodology which starts with the retrieval of data in an open-source dataset and ends with the assessment of several deep learning models such as CNNs and Vision Transformers (ViTs) on the task of multi-crop disease detection. This pipeline includes dataset preparation, image preprocessing and augmentation, model selection, model training and evaluation, and the comparative analysis of the model performance to benchmark with three crops that include maize, tomato and onion.

The initial stage was to obtain experimental data via the Mendeley public dataset repository (2024 edition), which has high-resolution images of crops on three key agricultural products. This dataset contains around 24,000 labeled images; these are healthy and diseased plants samples. The data set is further divided into three different sets namely maize (having 7 classes of diseases), tomato (having 12 classes of diseases which include fruit and leaf diseases) and onion (having 4 classes). Having these various types of plants provide a wide generalization capacity and the capability to study disease patterns in many crop families. This study did not involve the collection of the primary data since the dataset was not manually gathered. Yet, to provide training data diversity and to address the class imbalance issue, extensive set of image augmentation methods were used through transformations based on PIL (Python Imaging Library) and OpenCV. These were random rotation, flipping, brightness/contrast manipulation, zooming, and cropping, which were useful in simulating what a user might encounter in the real world and making the model more resilient to the severe conditions that the model might encounter in the real world. After this augmentation step, all images were scaled to the same input size usually 224x224 based on the needs of the model) and scaled to a range of 0-1 pixel values. This preprocessing step brought uniformity in all the inputs of the model and enhanced convergence during the training.

The following step of the methodology was a thorough model training pipeline based on the following six state-of-the-art CNN architectures: InceptionV3, InceptionResNetV2, DenseNet201, ResNet101, ResNet152V2, Xception. All models were fine-tuned through transfer learning where ImageNet weights were used as initial prototypes and the model was then extended with a lightweight classification head based on the target number of disease classes per crop. Train-validation-test split of 70:20:10 was followed in all the three datasets and training of models was conducted on each of the three crops (maize, tomato and onion) individually to test that the performance is consistent across domains. Standard classification metrics were applied in the evaluation of each trained model: accuracy, precision, recall, F1-score, and confusion matrix. Also, the amount of time used in inference to determine computational performance in practical, low-resource agricultural settings was measured.

Along with the experimentation based on the CNN, a Vision Transformer (ViT) model was also adopted to overcome the shortcomings of CNNs, including their limited receptive field and the inability to handle long-range spatial dependencies. Because it split the image into patches and used a self-attention mechanism to learn global relationships, the ViT model was an excellent fit to crops with disease symptoms potentially varying widely in location and size. The ViT model was also trained using the entire full dataset and compared to the CNN models to evaluate its performance in view of both accuracy and mode generalization across multiple crops. This comparison was used to not only test which architecture could be used to accomplish higher results overall, but which architecture provided a better trade-off between computational performance and accuracy of classification in the potential deployment of the architecture to a rural environment on a real-time basis.

Despite the fact that this given phase was not aimed at deployment and full-stack implementation, the lightweight character of the suggested models, especially the ViT-based approach, was taken into account with the references to future real-world applicability. The final goal is to come up with a solution that can potentially be incorporated into mobile or embedded platforms which can be made available to the farmers in remote locations. The approach forms the basis of the development of a scalable crop-agnostic deep learning-based disease detection system. It is also aimed to facilitate future growth into explainable AI (XAI) by absorbing interpretability methods like Grad-CAM, SHAP, and LIME, thus making the current model more trustworthy and transparent, decision-supporting tool in the agricultural industry.

1.5 Project Outcome

The given study makes a great contribution to the domain of agriculture and artificial intelligence, especially to the automated identification of the crop diseases that impact three staple food crops in South Asia: maize, tomato, and onion. Through incorporating the latest deep learning models with the feasibility of image classification in agriculture, the project shows how AI can be used to manage the diseases in the field, particularly in developing nations such as Bangladesh, where early detection of the disease can make a significant change in the crop production, the earnings of the farmer, and food security.

The project took off by using publicly available datasets of the three target crops. Both data sets were composed of high definition images of healthy and diseased plant leaves, which were taken in different environmental conditions. In total, the paper used a comprised dataset of over 24,000 images with more than 20 different classes of crop health and disease conditions. All these datasets were pre-selected, cleaned, and enriched with preprocessing and augmentation of the images available through the OpenCV and PIL libraries, which led to the well-balanced and diverse training sets. The most important finding of the research is the comparative assessment of several deep learning architectures, namely six of the most popular CNN models, i.e., InceptionV3, InceptionResNetV2, ResNet50, ResNet101, ResNet152, and Xception, and a self-designed Vision Transformer (ViT) model. The individual models were trained and fine-tuned on the maize, tomato and onion datasets and the performance was benchmarked to standard evaluation metrics including accuracy, precision, recall, F1-score, and inference time. The findings were in line with expectation as the proposed ViT model outperformed all three datasets in terms of either similar or better classification accuracy The ability to recognize the disease in the cross-crop manner makes the ViT model a significant point of the project result since it proves that it is possible to identify the disease in different crops with only one transformer-based architecture not requiring to construct separate models on each dataset. This makes the training overhead significantly less and introduces possibilities of implementing multi-crop diagnostic systems in limited resource settings like in a rural farm or an agricultural platform based on mobile devices. As opposed to standard CNNs, which emphasize local receptive fields, the self-attention component of the ViT architecture allowed the model to learn global spatial relationships in the case of plant leaf imagery. This became especially handy in identifying disease symptoms that do not exhibit regular patterns on the surface of leaves which is also a common phenomenon in diseases of crops in real life.

1.6 Organization of the Report

The report is organized into six detailed chapters, each prepared with a rational presentation of the research problem, analysis of related work, description of the used methodology, explanation of the experimental results, and conclusion with insight and future recommendations. This structure has been developed in order to make the study clear, coherent and technically deep so that the study is not only academically rigorous but also practically applicable to the domain of agricultural informatics.

Chapter 1: Introduction

This first chapter gives the backdrop to the study. It describes the extent of crop diseases worldwide and in the region on food security with the specific reference to maize, tomato and onion in Bangladesh and South Asia. Further, the motivation to detect the disease early and the description of deep learning models to solve this problem are also presented in the chapter. It also describes the purpose of the research, methodology and anticipated result of the study.

Chapter 2: Background

This chapter will provide a thorough literature review of studies that exist on the topic of detecting plant diseases through image classification. It discusses the major developments of Convolutional Neural Networks (CNNs) and transfer learning and the Vision Transformers (ViTs). Another point covered in the chapter is the available datasets, their shortcomings, and the way deep learning has been used in comparable agricultural issues. Moreover, it also notes the existing gap in the multi-crop detection systems and the importance of the trade-off between the performance and computational efficiency.

Chapter 3: Research Methodology

This chapter presents the approach taken in this research. It begins with a brief description of the dataset gathering on publicly available repositories, and goes into the specifics of preprocessing and augmentation (with OpenCV and PIL). The methodology then describes how six state-of-the-art CNN models and one of the custom ViT-based lightweight models are trained and assessed. It also presents a detailed description of the experimental set-up, model training approaches, model performance assessment parameters and the way the performance of each of the three models was evaluated in terms of robustness to the three crop types.

Chapter 4: Implementation and Results

This chapter gives details of how the proposed models were implemented and also a comparative analysis of the performance of each model based on accuracy, precision, recall, F1-score, and inference time. Evaluation is performed independently in maize, tomato and onion datasets, depicting the performance of the various architectures in different crop disease category. The output is visualized using confusion, graphs, and table which gives a clear picture on the performance of the model.

Chapter 5: Engineering Standards and Design Challenges

In this chapter, the engineering practices adhered to during the project such as adherence to software development standards and ethical concerns. It analyzes the scaling, efficiency as well as the rural deployment appropriateness of the system. It also draws attention to several important technical issues: class imbalance, computational constraints and interpretability of models and some of the techniques to address these issues. Financial, environmental and social implications of implementing such a system in agricultural areas are also examined.

Chapter 6: Conclusion

The final chapter is a general overview of the research and the possible applications of deep learning and ViT-based models in reliable multi-crop disease detection. It cogitates on how successful the project was in the development of performance, generalization, and cross-crop adaptability. The chapter also describes the shortcomings of the present work and gives recommendations to future works, including the incorporation of Explainable AI (XAI) algorithms and real-time implementation by mobile applications.

Chapter 2

Background

This chapter outlines the main concepts and new trends that are related to automated crop disease detection. It starts by introducing the concept of deep learning and how it can be used within the agricultural field and detecting crop diseases in particular. Particular attention is focused on Convolutional neural networks (CNNs), Vision transformers (ViT), and Explainable AI (XAI). An extensive literature survey is provided, which is concerned with deep learning in the classification of a disease in crops like maize, tomato, and onion. In its conclusion, the chapter highlights the research gaps especially in efficiency of the models, their interpretability and their flexibility in the deployment environments.

2.1 Introduction

The fundamental aim of the study is to come up with an automated and precise mechanism of identifying diseases in maize, tomato, and onion crops, which are amongst the three economically and nutritionally important agricultural commodities in Bangladesh and South Asia as a whole. These crops are very vulnerable to different bacteria, fungal and viral diseases which can devastatingly affect the quantity of yield and quality of crop, which can destroy food security and income of farmers. In rural and semi-urban agricultural societies, the unavailability of timely and precise diagnostic solutions usually results in the late intervention and extensive plant loss. Early and reliable detection of plant diseases in the presence of various environmental conditions is one of the most urgent problems of modern agriculture. Conventional modes of disease detection (e.g. expert inspection by eye or laboratory analysis) are slow, labor-intensive, expensive and not applicable to large-scale agriculture. These shortcomings have brought about the need to develop automated, cost effective and smart disease detection systems to assist in real time decision making and integrated pest management. In reaction to this need, deep learning has been observed to be an effective method in visual recognition in the agricultural field, especially in the classification of leaf diseases in crops. The models of deep learning, in particular Convolutional Neural Networks (CNNs) have proved to be highly successful in this area by training and extracting the complex visual features of diseased leaves, and not relying on manual feature engineering. The use of CNNs in plant disease studies has been extensive so far because CNNs can identify patterns of diseases like spots, discoloration, and unusual textures on their leaves.

Overall, this chapter is the basis upon which the knowledge of technical components and methodologies that are used in this research, will be shaped. It investigates the history of deep learning models in agriculture, reveals the noteworthy evolution and the significant progress in CNNs and ViTs, and it adds value to the growing significance of explainable AI. Survey of recent literature is also conducted to determine the existing solutions, including their success, and their shortcomings. Such understandings form the basis of inspiration to new contributions of this thesis, and it is highlighted in the following chapters of the methodology and the implementation of the thesis.

2.2 Literature Review

The application of deep learning architecture to detect plant diseases especially crops like tomatoes, maize and onion has gained popularity over the last ten years with various studies conducted on the topic. In the next section, a critique of current research carried out on this area is presented in terms of the models employed, their accuracies presented, as well as limitations to the different research articles. The comparative review can form a basis on which the current study can make a contribution, particularly the spheres of dataset diversity, cross-crop applicability, real-time deployment, and model interpretability.

2.2.1 CNN-Based Approaches

Several models have also managed to utilize Convolutional Neural Networks (CNNs) to predict crop diseases. As an example, Trivedi et al. (2021) using CNNs obtained a remarkable accuracy of 98.49%; nevertheless, their dataset size was small (3000 images) and only tomato diseases were considered, thus their findings are not likely to be transferred across crops [24]. Panigrahi et al. (2020) also utilized a CNN with ReLU activation and Adam optimizer and achieved 98.78% accuracy rate but the study did not place much stress on data augmentation which further doubted the model is sturdy in diverse environmental variation [25]. Another study by Olayiwola and Adejoju (2023) made an accuracy of 98.56% using a CNN developed using Keras and TensorFlow, but their analysis did not offer any variety of maize disease data [26]. Wenxia et al. (2021) used some inception and ResNet modules to form an enhanced CNN model, whose accuracy was 98%; however, the stack of modules may increase inferencing time within a real-time system [27].

2.2.2 Transfer Learning Approaches

Transfer learning has become popular in an attempt to overcome the lack of data in learning through the pre-trained models. Anandhakrishnan and Jaisakthi (2022) used Deep CNN with a train-test ratio of 60-40 and scored 98.4 percent accuracy, yet since the train-test ratio is low, one must consider the possibility of overfitting and the absence of cross-validation [28]. Thangaraj et al. (2021) did not report accuracy in transfer learning with different optimizers and their study cannot be applied to the general population because the diversity of the used datasets is insufficient to make inferences [29]. Another work by Kayaa et al. (2019) did not take the approach as specific as we did, since they examined the transfer learning with CNNs as well but provided limited insights into crop or disease-specific study [30]. Lopez-Martinez et al. (2024) tested a number of architectures- AlexNet, DenseNet, VGG16, SqueezeNet, MobileNetv2 and ResNet18 in onion field images captured by UAVs. Although they claimed 99% accuracy on the data set to which their method applied only on 3672 UAV images that was limited to situations where non-aerial or ground level imaging was required [31]. To investigate variations of CNN, Khan et al. (2024) tried a number of possible variations and found that ResNet50 (87.51% accuracy) worked the best, though the results of other CNN-based models (such as VGGNet or InceptionV3 were slightly worse [32]. This might be due to the complexity of the datasets or tuning of the model which is relatively lower than other study results.

2.2.3 Hybrid and Advanced CNN Models

Design of more complex models has been looked at as well. Attallah (2023) integrated traits of three small CNNs and combined them with the KNN and SVM classifier and returned exceptionally high classification statistics using both (KNN: 99.92%, SVM: 99.90%) [33]. But the fusion and hybrid pipeline might not work in low resource devices which is a major factor to consider in the rural agricultural context. Amin et al. (2022) presented a combination of EfficientNetB0 and DenseNet121 composing 98.56% of accuracy but having the downside of greater memory consumption and training overhead [34]. Sun et al. (2025) proposed a new CPAM module with a better EfficientNet with a performance of 98.32%, but the architecture might not be suitable to be used in real-time or edge environments owing to its complexity [35]. Nguyen et al. (2022) pre-processed VGG-19 with HSV-based segmentation to attain the test accuracy of 99.72%, yet this might decrease the feasibility of such a model in real-time, where raw data might be used as an input [36]. Guo et al. (2025) went the more elaborate path using stacked model where SEF-UNet, Res-UNet, SE Layer, and Error Level Analysis (ELA) were exercised. Though they approached a high accuracy rate of 92.62, their model was highly dependent on strict segmentation and consequently computationally expensive and scalability would be bad [12]. Meanwhile, a hybrid CNN-ViT model to classify maize leaf diseases was proposed in one of its other prominent works by Shandilya et al. (2025). The combination would enable CNN to be used to handle local features whereas ViT captures long-range dependencies. 5-fold cross-validation demonstrated that the model is generalizable with robust cross-dataset results of 99.15% accuracy [38].

2.2.4 Vision Transformers (ViT)

Whereas much of the reviewed research was dedicated to CNNs and other versions thereof, not many studies presented newer networks such as those of Vision Transformers (ViTs) which could provide better global attention as compared to CNNs. This disparity leaves an opening whereby future studies should attempt to integrate ViTs with explainable AI (XAI) to create better performance and interpretability.

ViT-SmartAgri is a ViT-based and smartphone-based solution to detect 10 of the 58 tomato leaf diseases presented by Barman et al. (2024) [39]. The study compared the ViT model with InceptionV3 and the control convention proves that ViT is superior with an accuracy of 90.99 percent testing accuracy hence can output such a model in real-time and in mobile-based plant health tracking. Fu et al. (2024), further applied ViT to recognizing images of crop pests also showing that it can provide leadership disease patterns even in non-obvious diseased area through block partitioning and self-attention [41]. They achieved a better performance using their improved ViT architecture that identified pests belonging to seven classes with high rates of accuracy therefore showing that their invention could effectively challenge manual or instrument-assisted methods. An in-depth study conducted by Ouamane et al. (2025) explored the idea of optimizing ViT so that improved results could be achieved on the PlantVillage dataset. They undertook a comprehensive analysis of different hyperparameters like patch size, embedding dimensions, depth, and attention heads. The most effective model was found to be exceptionally accurate at 99.77% with low computational overhead and memory usage and was superior to CNNs such as VGG19 and AlexNet, in both accuracy and performance [42].

Parez et al. (2023) described the GreenViT model that can address a CNN drawback the possibility of losing information in the case of pooling layers [43]. Due to this treatment of image patches as word embeddings, GreenViT was demonstrated to surpass state-of-the-art CNNs on benchmark datasets and, specifically, highlights the potential of transformers in precision agriculture. The study by Nishankar et al. (2025) responded to the necessity of having systematic analysis on ViT models on tomato leaf disease classification by means of the ViT-RoT framework [44]. This effort in benchmarking compared several ViT architectures and found that ConvNeXt-Small and Swin-Small were amongst the best. It emphasized on how to choose ViT models, which strike the right balance between accuracy, efficiency and scalability to the real world agriculture use. The work by Ali et al. (2025) forms a general review of the ViT models in plant disease classification numbering 55 classes. Their model had an accuracy of 89.3% proving the capability of transformers to work with large-scale, multi-class classification issues [45]. The paper has also identified ViTs as an effective alternative to conventional CNN-based networks, especially regarding versatility and performance Table 2.1.

Table 2.1: Summary of Literature Reviewed.

Reference	Model Used	Limitation	Accuracy (%)
Trivedi et al. (2021) [24]	CNN	Limited dataset (3000 images); focused only on tomato diseases	96.49
Anandhakrishnan & Jaisakthi (2022)	DCNN	Dataset split only 60-40; may lack cross-validation or generalization	95.4
Thangaraj et al. (2021)	Transfer Learning with DCNN	Evaluated with different optimizers but may lack dataset diversity	N/A
Kayaa et al. (2019)	Transfer Learning with CNN	General insight; lacks specific crop/disease implementation	N/A
López-Martínez et al. (2024)	AlexNet, DenseNet, VGG16, SqueezeNet, MobileNetV2, ResNet18	Dataset limited to UAV onion field images (3672 images)	97
Kavukcuoglu et al. (2010)	CNN (general foundation work)	Old foundational paper, not crop-specific	N/A
Huangfu et al. (2024)	ResNet50	None explicitly mentioned; general limitation: overfitting on small samples	94
Khan et al. (2024)	VGGNet, InceptionV3, ResNet50, InceptionResNetV2	ResNet50 best performer; others underperformed slightly	ResNet50: 87.51
Attallah (2023)	Fusion of 3 compact CNNs + KNN/SVM classifiers	Complexity in pipeline (fusion + hybrid feature selection); may not be suitable for low-resource devices	KNN: 99.92, SVM: 99.90
Sun et al. (2025)	CPAM + Improved	Complex architecture may	98.32%

	EfficientNet	increase computation time and limit deployment	
Guo et al. (2025) [12]	SEF-UNet + Res-UNet + SE Layer + ELA	High dependency on accurate segmentation increases preprocessing complexity	92.62%
Panigrahi et al. (2022) [25]	CNN with ReLU + Adam Optimizer	May not generalize well across diverse environmental conditions without augmentation	95.78%
Olayiwola & Adejoju (2023) [26]	CNN model using Keras on TensorFlow	Limited evaluation on diverse maize disease datasets	95.56%
Wenxia et al. (2021)	Improved CNN with Inception + ResNet modules	Increased model complexity due to module stacking may slow down inference	96%
Amin et al. (2022)	EfficientNetB0 + DenseNet121	Feature fusion leads to increased memory usage and training overhead	94.56%

2.2.5 Similar Application

Early and proper diagnosis of the diseases affecting crops is very crucial in reducing the wastage of crops and increasing crop yield. In the previous ten years, deep learning, computer vision, web/mobile technologies have become more and more used by researchers to create working solutions to allow diagnosing diseases at early stages and monitor crop health. The present section is a summarization of notable existing applications, both of the research and of actual deployment, which show a very close fit to the objectives of this thesis.

DeepCrop developed by Islam et al. (2023) with the use of ResNet-50 model is a web application used to diagnose various leaf diseases through the use of leaves [46]. It tested the CNN, VGG-16 VGG-19 and ResNet-50 on PlantVillage dataset and concluded that ResNet-50 was best with an accuracy of 98.98 percent hence it was incorporated into the deployed web system. The app enables farmers to share leaf pictures and obtain the classification of disease in real-time, therefore, allowing timely, affordable control. Nandhini et al. (2018) introduced the compressed sensing and support vector machines (SVM)-based Web-enabled Disease Detection System (WEDDS). Their solution contains cloud-based analysis in order to decrease weight on local processing and demonstrated classification accuracy of 98.4% [47]. Kheir et al. (2025) proposed a hybrid deep learning web environment with MobileViTV2 and EfficientNet-B7 on a disease with 4 classes of healthy, rust, scab, and multi-disease. The model MobileViTv2 showed the highest accuracy rate of 94% and the system offered a web-based real-time interface to make a diagnosis [48]. They also read out confidence scores of identifying disease in their platform that was commended as being highly user-friendly and scalable.

Manasa (2023) created a mobile application based on the CNN, via which farmers can take photos of diseased tea, and the system returns the respective classification with the confidence level [49]. Essentially, Obadaarachchi et al. (2023) developed a bespoke CNN which demonstrated better functionality than ResNet50V2 and has a 94% get-real-time detection and recommendation [50]. Tembhurne et al. (2023) fine-tuned the MobileNet on an image dataset with 64 disease classes across 22 crop types containing 12,318 images with the use of the Keras Tuner. Their system had an accuracy of 95.94% and hence being among the most generalizable model across crop species [51]. Last but not least, Ahmed & Reddy (2023) also has CNN trained on a huge database of 96,206 images of 38 diseases. The Android app allows users to diagnose 14 different types of crops with overall accuracy of 94 percent and provide actionable insight to help the user cut fertilizer waste and enhance production.

Table 2.2: Similar applications

Reference	**Models**	**Application Type**	**Accuracy**	**Highlights**
Islam et al. (2023)	ResNet-50	Web App	94.78%	DeepCrop for real-time crop disease diagnosis
Nandhini et al. (2018)	SVM + Compressed Sensing	Web System	96.40%	Lightweight, Raspberry Pi-based system
Kheir et al. (2025)	MobileViTv2, EfficientNet-B7	Web App	94%,	Real-time leaf disease classification with high generalization
Jadesha et al. (2025)	Random Forest	Web App	97%	Predictive modeling using weather and crop cycles
Hang et al. (2023)	VGG16, VGG19, Xception, etc. (Ensemble)	Mobile App	93.80%	Fuzzy Choquet ensemble with high robustness
Obadaarachchi et al. (2023)	Custom CNN	Real-Time Mobile System	94%	Outperforms ResNet50V2; real-time treatment recommendations
Li et al. (2023)	MobileNetV3 + CA	Mobile App	95.88%	Improved architecture for tea leaf disease and insect diagnosis
Lanjewar et al. (2023)	CNN + ResNet50, NASNet	Cloud Platform (PaaS)	97%	Cloud-deployed, mobile-accessible disease prediction
Tembhurne et al. (2023)	MobileNet (Keras Tuned)	Unified Mobile App	95.94%	64 disease classes, 22 crop types
Ahmed & Reddy (2023)	CNN	Android App	94%	38 disease categories across 14 crops using 96,206 images

2.3 Gap Analysis

Deep learning applications in diagnosing plant diseases have seen a tremendous growth with a surge of the number of web and mobile-based apps, developed using CNNs, transfer learning, hybrid systems and lately, the so-called, Vision Transformers (ViTs). Although the level of improvement is impressive, further evaluation of the level of existing body of work leaves essential gaps and areas unexplored, which this study strives to address.

Table 2.3 Gap analysis summary table

References	**XAI**	**Multi-class crops**	**Generalization across multiple diseases**	**Model deployment**	**Visual diagnostic**
Islam et al. (2023)	No	Yes	No	No	No
Kheir et al. (2025)	No	No	Yes	No	No
Hang et al. (2023)	No	No	No	No	No
Barman et al. (2024) [39]	Yes	Yes	Yes	No	No
Fu et al. (2024)	Yes	Yes	Yes	No	No
Ouamane et al. (2025)	No	Yes	No	No	Yes
Parez et al. (2023)	No	Yes	No	No	No
Nishankar et al. (2025)	No	Yes	Yes	No	No
Shandilya et al. (2025)	No	No	No	No	Yes
Ali et al. (2025)	No	No	No	No	No
This Study	Yes	Yes	Yes	Yes	Yes

2.4 Summary

Here, the existing state of plant disease diagnoses in various crops, such as maize, tomato, and onion, was examined through the lens of deep learning-based approaches, namely CNNs, transfer learning, hybrid model, and the Vision Transformer-based model or the ViTs. The literature showed that there has been massive improvement in accuracy and usage of disease detection systems through web and mobile systems. CNN models and transfer learning networks showed great success, which is especially true when applied in the fine-tuning to a particular crop. Nevertheless, most of the available solutions continue to be crop-specific, which does not give them tangible scope in various forms of agricultural settings. Lightweight models and hybrid model architectures have opened the door to better scalability, but computation and preprocessing demands continue to be a factor dragging behind the ability to run on low-resource or mobile devices. Besides, even though Vision Transformers are promising to exploit global image features, most existing ViT-based models lack explainability and the capability of optimization on edge devices. Important explainable AI (XAI) methods such as LIME, SHAP, and Grad-CAM that are essential to establish trust and transparency of a system with the user are not widely used in these applications. Altogether, current systems are restricted by the factors of multi-crop generalizability, real-time deployment, interpretability, and user-centered design, despite the rapid progress. This review shows that there is a necessity of a unified, lightweight, explainable multi-crop plant disease diagnosis system with an adequate balance between accuracy, efficiency and usability to be applicable to real world agriculture. The proposed study will attempt to overcome these research gaps by devising a scalable, interpretable, and robust framework capable of sustaining decision-making requisites in the real world in the context of diverse crop disease detection.

.

Chapter 3

Research Methodology

This study has given the systematic approach to create an explainable, lightweight, and accurate deep learning system, used to detect diseases in the maize, tomato, and onion leaves. The methodology is composed of a sequential flow, which is started by data collection, is then followed by a preprocessing procedure, a model development and evaluation process, and finally, explanation methods integration. The end objective is to make the outcome model generalizable, interpretable, and applicable in the real world in agriculture.

3.1 Methodology

3.1.1 Overview

The design of the research undertaken in this project is with reference to the organized rubric of establishing sound and explainable deep learning pipeline regarding multi-crop plants disease judgments. The system under development is made in steps to achieve accuracy, efficiency, and capacity to work across all crop types of interest namely maize, tomato and onion. The initial step of the methodology is data collection via publicly open data. These data contain not only labeled images of both healthy and diseased leaves of all the three crops, but also provide a wide-ranging and substantial training basis. Pictures are posed differently in terms of their light, backgrounds and the level of disease to resemble real situation in the field. After collecting data, preprocessing is done to make sure the information fed to deep learning models can be used in training deep learning models. All the images are scaled to the same resolution (224224), meanly standardized to ensure that pixel values lie within a certain range, and divisible into training, validation and testing sets based on a 70 20 10 ratio. The main experimental part will consist of such steps as training and testing several deep learning models. This comprises several transfer learning using a Vision Transformer (ViT) from scratch which is the suggested ultimate unified architecture. Each model is trained separately on the three crop datasets to see what crop-specific performance looks like and then eventually evaluated together in order to determine generalizability. The standard metrics to be used in measuring performance are accuracy, precision, recall, F1 score, and confusion matrix. In a bid to enhance explaining, the tools of Explainable AI (XAI) techniques like SHAP, LIME, and attention heat maps, are used. Also, pseudo-segmentation methods are included, to make predictions in the input images more understandable to farmers and other agricultural specialists. Even though mobile deployment is not used in this initial study, the resultant model will be compatible and lightweight enough to be used on instances of low-resource or mobile devices in subsequent versions.

3.1.2 Proposed Methodology

In this study, the goal is to create an efficient deep learning system to be used in multi-crop disease detection consisting of maize, tomato, and onion with potential to be explainable. It starts with the gathering of publicly accessible sets of images of healthy and diseased leaves of these crops. The obtained data is preprocessed: it is resized to a standard resolution and normalized. Methods of PIL and OpenCV data augmentation are implemented to enhance the performance of the model and manage the imbalanced classes. The dataset will then be partitioned into three different sets; training, validation, and testing. Various models are shaped and tested, such as a custom lightweight CNN and various pre-trained networks meant to accomplish feature extraction and its classification. The training includes a lot of hyperparameter adjustments using methods of early stopping and learning rate schedule to avoid overfitting. Further, hyperparameter optimization of the ablation experiments are run on the maize dataset for factors including optimizer, activation function, size of input image, patch dimension and transformer layer architecture. The most successful model setup is then tested upon the tomato and onion datasets to be tested. The measurement of performance is done on basis of such names as accuracy, precision, recall, F1-score, and confusion matrix. Explainable AI techniques including SHAP, LIME and attention heatmaps are applied to explain why the leaf images were classified in this way by indicating regions of interest within the leaf image that are associated with the decisions. Lastly, comparison is written against what exists in literature to show the effectiveness and strength of the model.

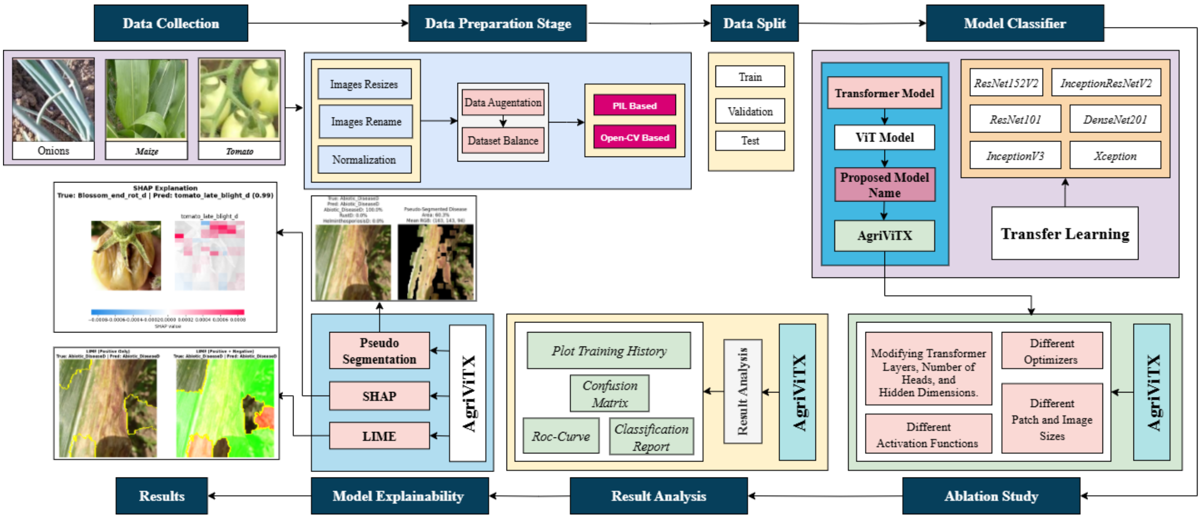


Figure 3.1: Flowchart of the Proposed Methodology

The model is also applicable in fields through future deployment on mobile or edge devices, owing to its lightweight state. In this study, the goal is to create an efficient deep learning system to be used in multi-crop disease detection consisting of maize, tomato, and onion with potential to be explainable. It starts with the gathering of publicly accessible sets of images of healthy and diseased leaves of these crops. The obtained data is preprocessed: it is resized to a standard resolution and normalized. Methods of PIL and OpenCV data augmentation are implemented to enhance the performance of the model and manage the imbalanced classes. The dataset will then be partitioned into three different sets; training, validation, and testing. Various models are shaped and tested, such as a custom lightweight CNN and various pre-trained networks meant to accomplish feature extraction and its classification. The training includes a lot of hyperparameter adjustments using methods of early stopping and learning rate schedule to avoid overfitting. Further, hyperparameter optimization of the ablation experiments are run on the maize dataset for factors including optimizer, activation function, size of input image, patch dimension and transformer layer architecture. The most successful model setup is then tested upon the tomato and onion datasets to be tested. The measurement of performance is done on basis of such names as accuracy, precision, recall, F1-score, and confusion matrix. Explainable AI techniques including SHAP, LIME and attention heatmaps are applied to explain why the leaf images were classified in this way by indicating regions of interest within the leaf image that are associated with the decisions. Lastly, comparison is written against what exists in literature to show the effectiveness and strength of the model. The model is also applicable in fields through future deployment on mobile or edge devices, owing to its lightweight state.

3.1.3 Functional and Nonfunctional Requirements

The model under consideration believes in the fundamental functional as well as non-functional requirements as to achieve conformity with respect to consistency, usability, and functionality within real-life agricultural environments. The existence of functional requirements is used to identify the main actions that the system should perform whereas the non-functional requirements should identify the quality specifications like performance, usability, and security.

Functional Requirements

The Functional requirements give the essential functionality and user interactions of the multi-crop disease detection system:

- **Image Upload:** This system provides an option to users to upload an image of maize, tomato, or onion leaves, i.e., by capturing an image through the camera or choosing an image in the gallery on the device.
- **Preprocessing of images:** The uploaded images are further scaled, normalized, and augmented (optional) to be ready to be classified on diseases.
- **Disease Classification:** This system is able to interpret the input image into a set of predetermined classes (i.e., various disease types and healthy) particular to maize, tomato, and onion based on the trained lightweight CNN and transfer learning systems.
- **Model Selection and Inference:** The system allows using multiple CNN architectures and a custom lightweight model that is optimized by hyperparameter tuning and ablation.
- **Output:** Prediction (e.g.: "Leaf Blight - 94.3% Confidence")
- **Explainability Module:** Explainable AI parameters like SHAP, LIME, and attention heatmaps will also be integrated with the system to help the user understand why a certain place in the image affects its classification and gain a sense of trust through it.
- **Comparative Analysis:** Allows the comparison of the performance of the proposed model with existing state-of-the-art models with the metrics and plots.
- **User Feedback:** The system has facilities that allow the user to later give feedback on the accuracy of the prediction and the usability of the application and this feedback will help it improve the model.

- **Online Prediction Capability:** The model can be later used to do online inference with mobile devices, and people in areas with low connectivity can use it to diagnose the disease.
- **User Notifications:** User can get visual or text status of uploading, progress and error notification.

Nonfunctional Requirements

The non-functional requirements ensure the quality, reliability and the ability of the system to operate as follows:

- **Performance:** The inference results: The inference results need to be delivered within 3-5 seconds on mobile environments as well as web environments.

- **Accuracy:** The model is expected to have balanced accuracy and precision on all the crop sets with at least 97 percent classification accuracy.

- **Usability:** The interface should be easy and responsive and anybody, with little knowledge of the technical aspect should be able to use that interface with ease.

- **Scalability:** The framework must be able to support future work, such as in support of additional types of crops, types of diseases, and more advanced ensemble or transformer-based models.

- **Maintainability:** The modular design should support full version control, debugging and updating to newer stable, without incurring a lot of downtime.

- **Reliability:** The possibility of the system performing similarly in varied environmental conditions and on the varied operating systems should result in reliable forecasting.

- **Security and Privacy:** The images uploaded by the users and their personal data have to be securely encrypted during storage and transfer in accordance to privacy policies.

- **Resource Efficiency:** Low-end mobile can trigger with limited CPU and memory capacity thus the model should be resource efficient.

3.1.4 Data Flow Diagram

The Level 1 Data Flow Diagram shows the main elements and transitions of information in Crop Leaf Disease Classification System specifically in maize, tomato and onion crops. In this, the User starts by uploading a raw picture of a leaf of one of the target crops. The preprocessing stage is where the image is initially processed by the Image Input Module and passed to the Preprocessing Module where it is resized, normalized, and placed to undergo the prediction or inference using a model. Then, the prepossessed image is transmitted through the Model Inference Module, where the disease is classified by specific lightweight CNN, which is a proprietary model. A centralized Model Repository we retrieves the model. The result of the prediction is subsequently fed into XAI Module which makes the visual explanation of the classification outcome using different techniques like Grad-CAM, SHAP, and LIME. The explanation is kept in Explanation Output Store. Last, the detected type of the disease (e.g. Tomato Early Blight, Maize Leaf Spot, Onion Downy Mildew) along with its graphical description is displayed to the User through the Result Display Module.

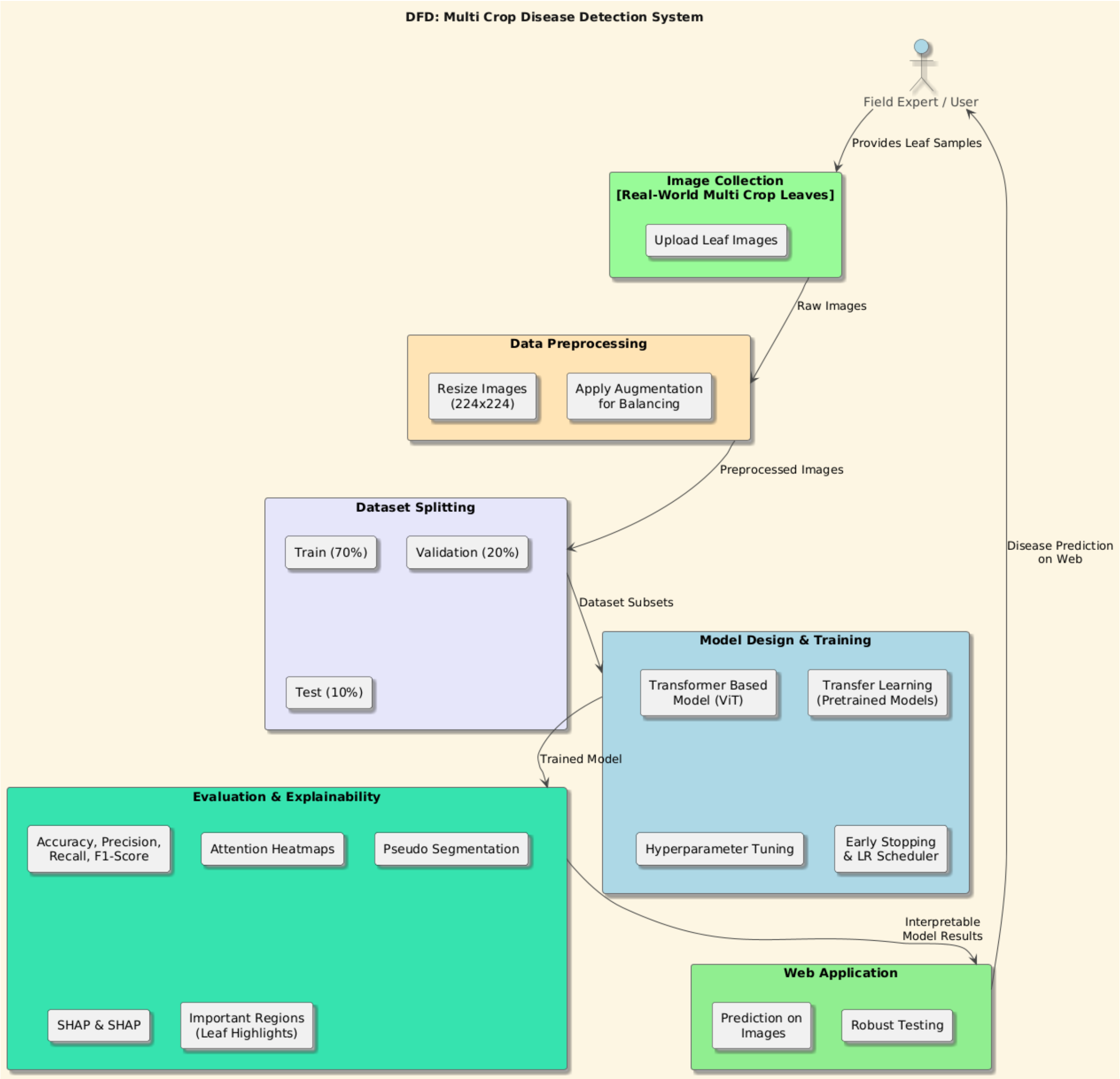
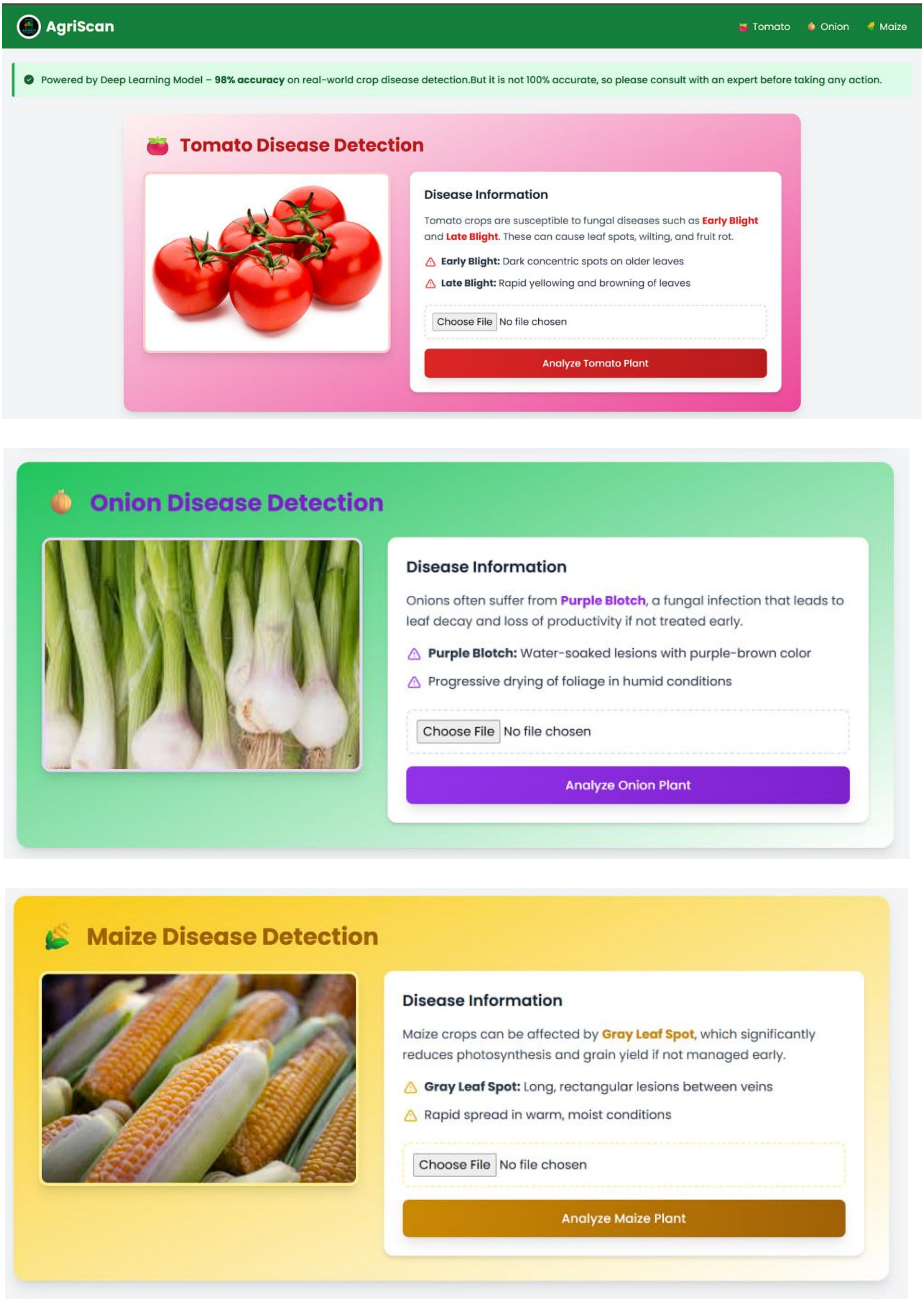


Figure 3.2: Data Flow Diagram of The Proposed System

3.1.5 UI Design

The proposed system has a user interface (UI) designed to ensure simplicity and accessibility for users. The web-based design provides an intuitive layout with clearly defined sections for uploading and analyzing images. Specifically, the interface allows users to upload images of tomato, onion, and maize leaves. Once an image is uploaded, the system automatically processes it and displays whether the plant is healthy or affected by any disease. This interactive design not only enhances user experience but also validates the results in real-time, ensuring that users receive instant feedback regarding the health status of the plant. The proposed system has a user interface design (UI) as follows:



Powered by Deep Learning Model – **98% accuracy** on real-world crop disease detection.But it is not 100% accurate, so please consult with an expert before taking any action.

Tomato Disease Detection

Disease Information

Tomato crops are susceptible to fungal diseases such as **Early Blight** and **Late Blight**. These can cause leaf spots, wilting, and fruit rot.

⚠ **Early Blight:** Dark concentric spots on older leaves

⚠ **Late Blight:** Rapid yellowing and browning of leaves

Onion Disease Detection

Disease Information

Onions often suffer from **Purple Blotch**, a fungal infection that leads to leaf decay and loss of productivity if not treated early.

⚠ **Purple Blotch:** Water-soaked lesions with purple-brown color

⚠ Progressive drying of foliage in humid conditions

Maize Disease Detection

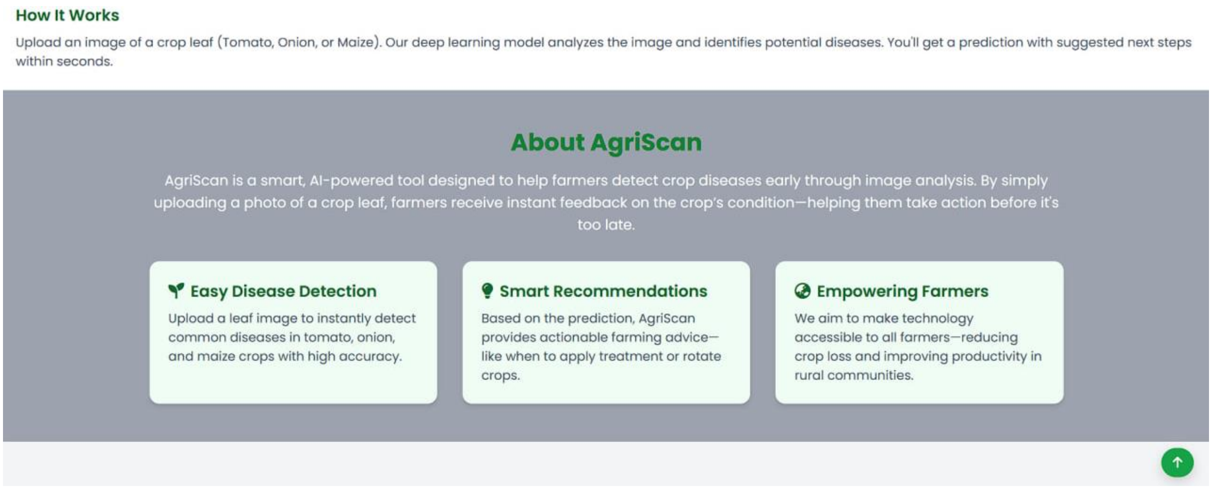
Disease Information

Maize crops can be affected by **Gray Leaf Spot**, which significantly reduces photosynthesis and grain yield if not managed early.

⚠ **Gray Leaf Spot:** Long, rectangular lesions between veins

⚠ Rapid spread in warm, moist conditions

Figure 3.3: Homepage of AgriScan



How It Works

Upload an image of a crop leaf (Tomato, Onion, or Maize). Our deep learning model analyzes the image and identifies potential diseases. You'll get a prediction with suggested next steps within seconds.

About AgriScan

AgriScan is a smart, AI-powered tool designed to help farmers detect crop diseases early through image analysis. By simply uploading a photo of a crop leaf, farmers receive instant feedback on the crop's condition—helping them take action before it's too late.

Easy Disease Detection

Upload a leaf image to instantly detect common diseases in tomato, onion, and maize crops with high accuracy.

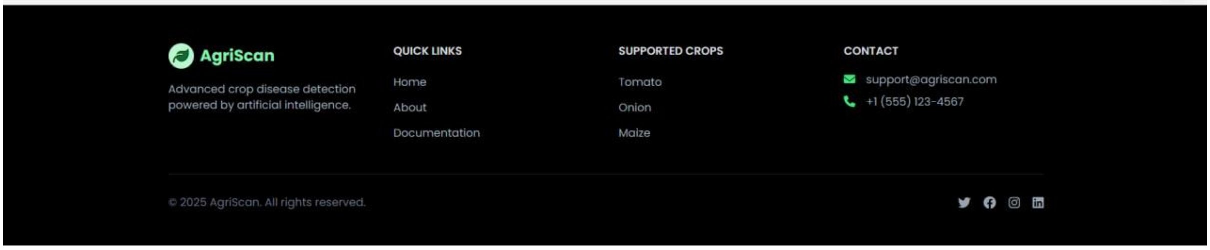
Smart Recommendations

Based on the prediction, AgriScan provides actionable farming advice—like when to apply treatment or rotate crops.

Empowering Farmers

We aim to make technology accessible to all farmers—reducing crop loss and improving productivity in rural communities.

Figure 3.4: About Section



AgriScan

Advanced crop disease detection powered by artificial intelligence.

QUICK LINKS

Home
About
Documentation

SUPPORTED CROPS

Tomato
Onion
Maize

CONTACT

support@agriscan.com
+1 (555) 123-4567

© 2025 AgriScan. All rights reserved.

Figure 3.5: Footer Section

3.2 Detailed Methodology and Design

The design of the experimental pipeline to classify multi-crop disease in leaves is finely organized in a number of consecutive steps, guarantees the performance, interpretability and viability of the models in practice. It begins with obtaining images of high-quality leaves of three types of crops that are maize, tomato, and onion with the rigorous preprocessing and augmentation procedures to normalize and diversify the data. Its main three stages of experiments are as follows: the research completes its development with an interpretation of models and deployment to practice with the use of cross-platform tools. Each step is intentionally created in order to derive knowledge, maximize efficiency and improve explainability. The six CNN architecture with transfer learning (TL) used at the first stage include ResNet152V2, DenseNet201, InceptionV3, ResNet101, InceptionResNetV2, and Xception. These models, having been pre-trained on ImageNet, will get fine-tuned on pre-processed crop-specific leaf datasets. This is intended to measure the number of diseases that were classified accurately by the traditional CNN architectures in the three target crops.

The second phase is the training of a new model, its implementation is based on Vision Transformer (ViT), which is trained a new and then tested. That way, the study delves into the transformer-based architecture in comparison to CNN. Several ablation experiments are carried out, like playing around with patch size, image resolutions, activation, optimizers, and hidden layers structures in order to establish optimal architecture parameters. At the third step, the best configuration of ViT model is retrained and tested on the whole combined data (training all three crops: maize, tomato, onion) to test its scalability, generalization and coping with the multiclass, multi-crop problem. During the fourth phase, we combine the ViT-based model with Explainable AI (XAI) techniques SHAP, LIME and attention based heatmaps. Such methods offer interpretable and clear visualizations which point out the most influential parts of each leaf image in the resulting predictions. The last step of optimization ViT model is directly deployed to the corresponding web interfaces that allow right away disease prediction via browsers This step will confirm that the system is not only accurate and explainable, it is scalable, accessible, and appropriate to use in resource-limited regions.

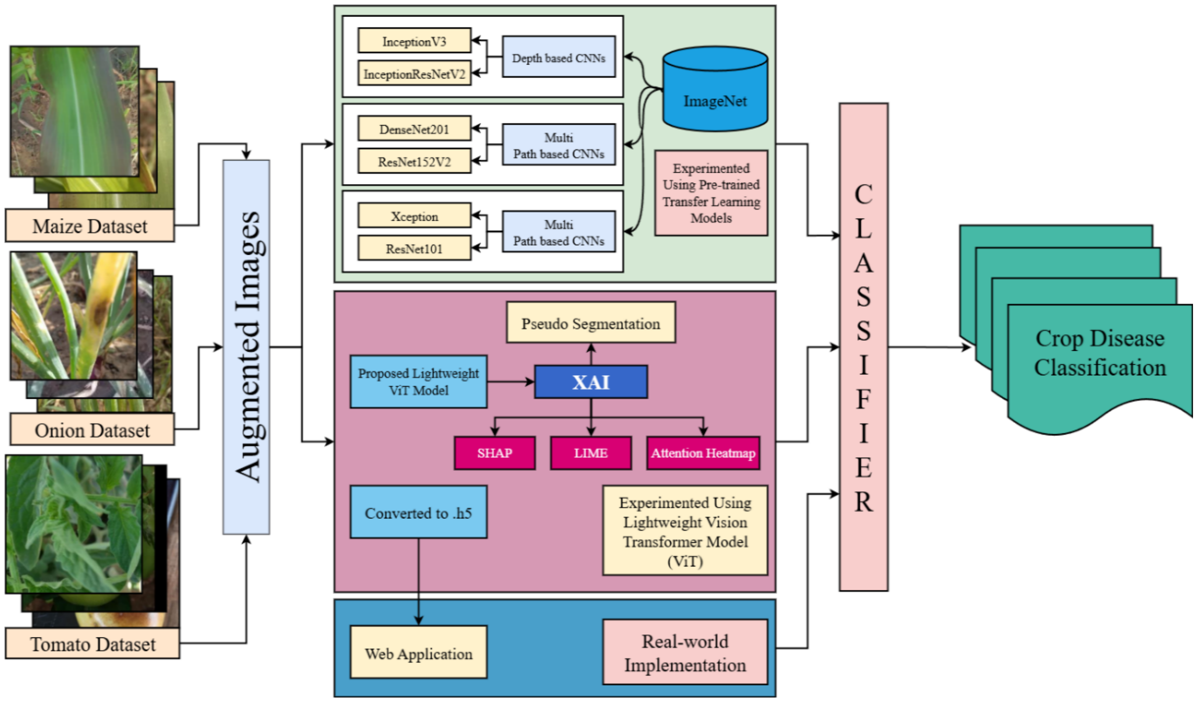


Figure 3.6: Experimental workflow of this study.

3.2.1 Dataset Description

The research work was performed using a publicly accessible dataset called TOM2024 that was specifically intended to address the issue of agricultural image classification [52]. The dataset includes 25844 raw pictures generated out of three key crops such as tomato, onion and maize. They were taken in different environmental conditions and divided into 30 different classes that depicted the different crop diseases and plant conditions. Nonetheless, this study selected only 23 classes out of these 30 classes. The rest of the classes that significantly constituted crop pests, were deliberately left out, since classification proposed in this study was limited to crop and leaf diseases alone, and any data relating to pests was considered irrelevant to research scope of this paper. The selection of these classes of disease was done critically because the selected classes of disease directly correspond to the symptoms such as leaves and crop, which coincide with the aim of this research perfectly.

Figure 3.7: Sample Images from Each Class of the **Tomato Dataset**

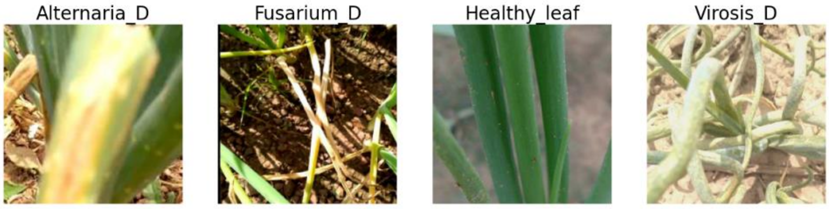


Figure 3.8: Sample Images from Each Class of the **Onion Dataset**

Figure 3.9: Sample Images from Each Class of the **Maize Dataset**

3.2.2 Data Preparation Stage

In order to ensure that our model is reliable and handle the issue of class imbalance, we performed a complete series of the preprocessing and data augmentation operations in the raw maize leaf disease image datasets. This was aimed at increasing the classes to images leading to better generalization of the model in training. We applied a combination of PIL and OpenCV transformations here all carried out in Python.

PIL-based Augmentations

We applied various augmentations of PIL to make the training data much more real and resembling the kind of thing you would see in the field. Brightness was varied randomly between 0.7 and 1.3 in order to resemble the effects of varying natural light in the outside world. Various levels of contrast were processed inside the same range using the values provided by Wu et al. (2021), which has contributed to retaining significant details of the leaves and providing some handy differentiation at the same time. We also adjusted the color a little- again to a point between 0.7 and 1.3- on the recommendation of Too et al. (2019) which found that such a change keeps images looking natural without affecting any of the biological characteristics of the leaves or fruits. The augmented images were stored with the full information of their transformations to allow the traceability as suggested by Ghazi et al. (2019), such as in the filename (i.e., augmented_pil_0_b0.92_c1.14_col1.05.jpg). As in the case done by Mohanty et al. (2016) we ensured that each image was visually checked prior to finalizing the dataset to avoid glitches and other strange artifacts. As soon as something did not seem just right, we changed the parameters. This aided in ensuring that only quality and realistic samples remain that may actually retain the disease characteristics which enhances the reliability within the model.

Figure 3.10: Augmentation Technique Visualization

OpenCV-based Augmentations

To improve model robustness and generalization, an augmentation of the original image was performed with OpenCV of multiple manipulations of space and visual distortion typical of a real agricultural environment. The said augmentation strategies involved centre cropping (crop sizes of 0.8 and 0.6) and scaling (zoom-in) (with a scale factor of 1.1 and 1.3). Horizontal and vertical flipping were performed so as to create a mirrored leaf orientations. Rotational transformations have occurred at an angle of 15 degrees, 30 degrees, 15 degrees, and 30 degrees and shifting (translation) saw movements in the images of 10 pixels in the x direction or y direction. Also, zoom augmentations with 1.2, 1.5 and 1.8 factors were added to imitate other camera distances. To simulate the visual distortions, all types of blur were applied: Gaussian blur, median blur, and motion blur, with kernel radiuses of 3, 5 and 7. To make the process traceable and clear, it was important to save every augmented image with certain descriptive name that signifies the transformation used-such as names of the files like rotated_15_12.jpg or gaussian_blur_5_34.jpg were used.

Table 3.1: Dataset Distribution After Augmentation

Crop	Class Count	Original Images	Augmented Training Images	Augmented Validation Images	Augmented Test Images	Total Augmented Images
Maize	7	3,509	10,780	3,080	1,540	15400
Onion	5	2,165	1,960	560	280	2800
Tomato	12	4,152	13,440	3,840	1,920	37400
Total	24	9826	26,180	7,480	3,740	55600

3.2.3 Model Selection

This paper chose a wide range of CNN architectures to see how they perform as classifiers of crop diseases. The models selected belong to various classes of the deeper taxonomy of CNN architecture (Figure 3.11) [53]. The models are categorized into the, ResNet152V2, ResNet101 and DenseNet201 into the Depth based CNNs, InceptionV3, and InceptionResNet152V2 into the Multi Path based CNNs, Xception into the Width based Multi-Connection.

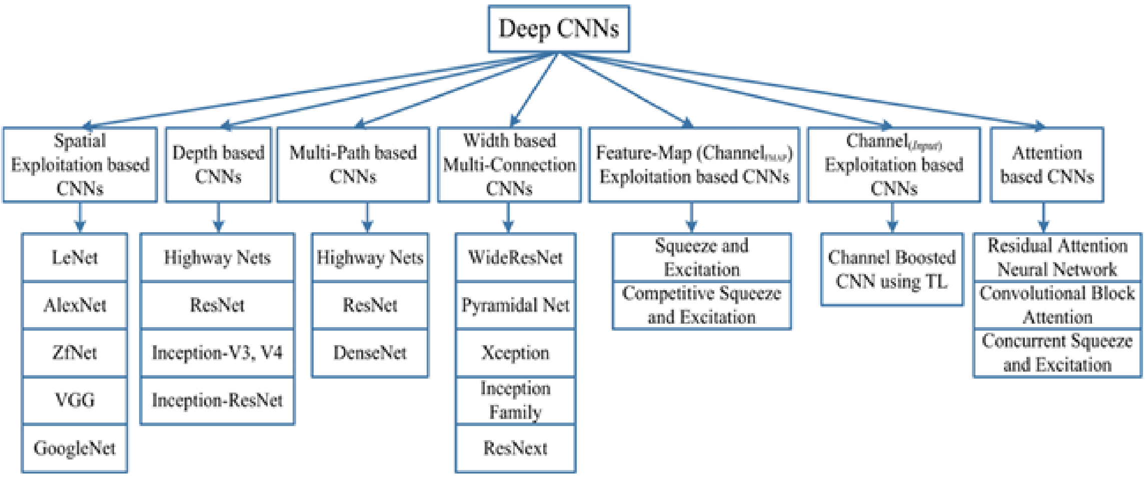


Figure 3.11: Taxonomy of deep CNN architectures

a) InceptionV3

InceptionV3 is an extended (by architectural refinements) deep CNN network of the Inception model optimized internal network performance in terms of structural innovations like factorized convolutions, asymmetric convolutions, and auxiliary classifiers. The methods enable one to extract features in a multi-scale manner efficiently and lower computer power requirements. Reduction of the size of the grids also addresses the number of parameters to a great degree without compromising the accuracy. The architecture of InceptionV3 ensures rich hierarchical representations can be acquired and this will help in classifying various categories of diseases in crop leaves [54]. InceptionV3 architecture is illustrated in figure 3.12.

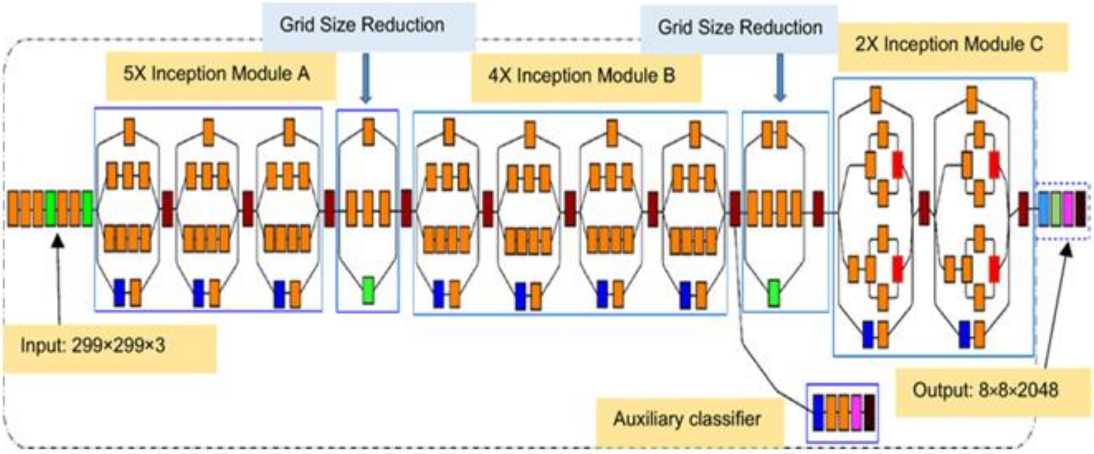


Figure 3.12: Architecture of InceptionV3 (source: Google)

b) InceptionResNetV2

InceptionResNetV2 is an architecture based on hybrid combination of factors. It is an architecture that runs on multi-scale feature extraction of Inception modules and residuals of ResNet. The model will seek to take advantage of the strengths of both Inception and ResNet architectural designs; namely width and parallel processing and depth and gradient stability, respectively [55].

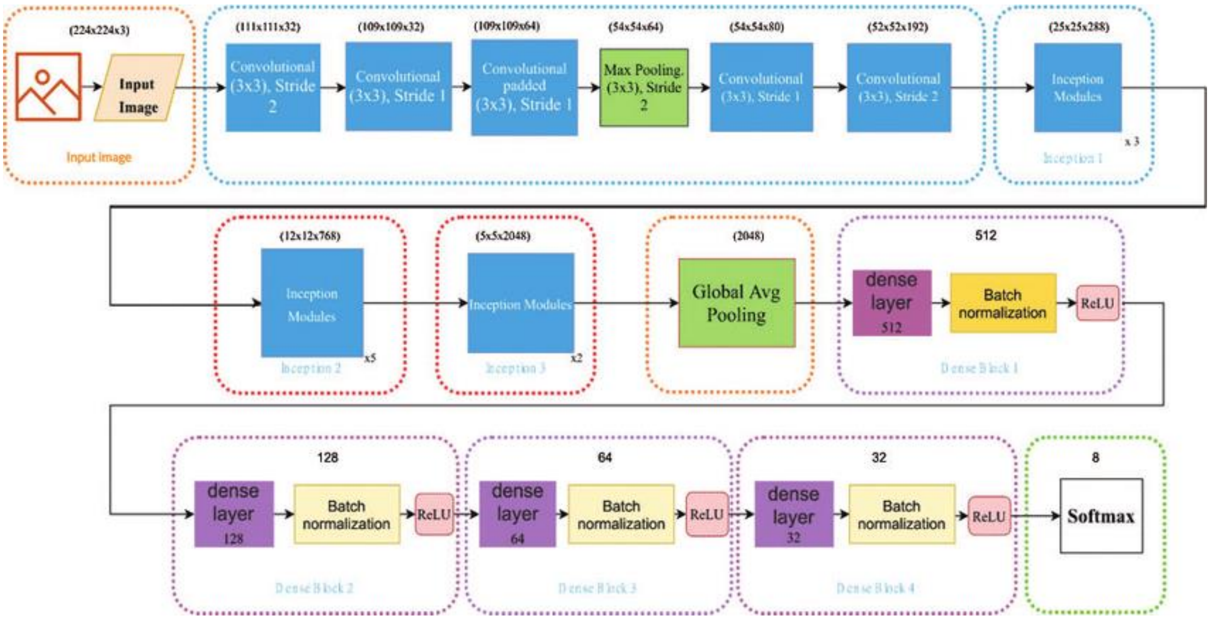


Figure 3.13: Architecture of InceptionResNetV2 (Source: Google)

This architecture consists of three larger sections, namely, the stem, which preprocesses the input image into a base layer of feature map; the Inception-ResNet-A, B, C modules which operate on increasingly deeper feature representations and the reduction modules which reduce the spatial dimensions whilst increasing feature depth. The best trait of InceptionResNetV2 is identifying injured differences in the disease classes in the leaf images since it can extract the complicated spatial hierarchies. Figure 3.13 shows the entire architecture.

c) DenseNet201

DenseNet201 is a deep convolutional neural network that adopts dense connectivity i.e. every layer is connected to all of the previous and succeeding layers, accepting as input the feature maps of all the previous layers. Such architecture can assist in addressing the problem of reducing the vanishing gradient and enhances feature propagation and reuse that helps to enhance learning efficiency and minimizes redundancy. The network consists of numerous dense blocks with interconnections between each other provided by transition layers. Every dense block has several convolutional layers directly coming into contact with all other layers in a feed-forward format. This gives a more efficient and compact model with fewer parameters than traditional architectures of comparable depth. The deep hierarchical feature extraction and better gradient flow of DenseNet201 enabled it to fulfill complicated classification tasks which include multi-disease identification in crop leaves. It has good performance in transfer learning environments which are an indication that it has the ability to generalize well across domains [56]. The architectural structure of DenseNet201 is shown in Figure 3.14.

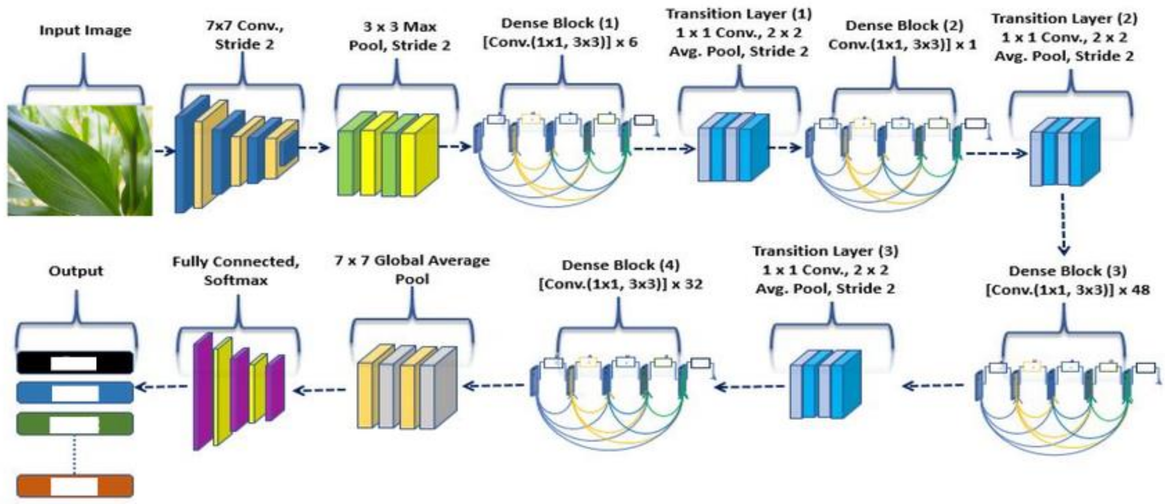


Figure 3.14: Architecture of DenseNet201 (Source: Google)

d) ResNet152V2

ResNet152V2 is an extremely deep residual network with 152 layers, to solve the degradation issue of deep networks, an identity shortcut connection (another name is residual connection) was introduced. Such links permit residual mappings to be learned by the network, which makes it easy to train the network although its depth is much larger. V2 variation of ResNet provides batch normalization before the activation (pre-activation) which enhances the process of optimization as well as minimizing internal covariate shift. The model consists of the bottleneck residual blocks which assist in minimizing the amount of parameters and keeping the depth. The depth and structure of the ResNet152V2 is capable of extracting fine and abstract features that will be needed to categorize the vast amount of disease classes that are imaged in the leaves. It delivers good transfer learning results in transfer learning tasks especially when it has been trained on agricultural datasets. Figure 3.15 shows the architecture of ResNet152V2 [57].

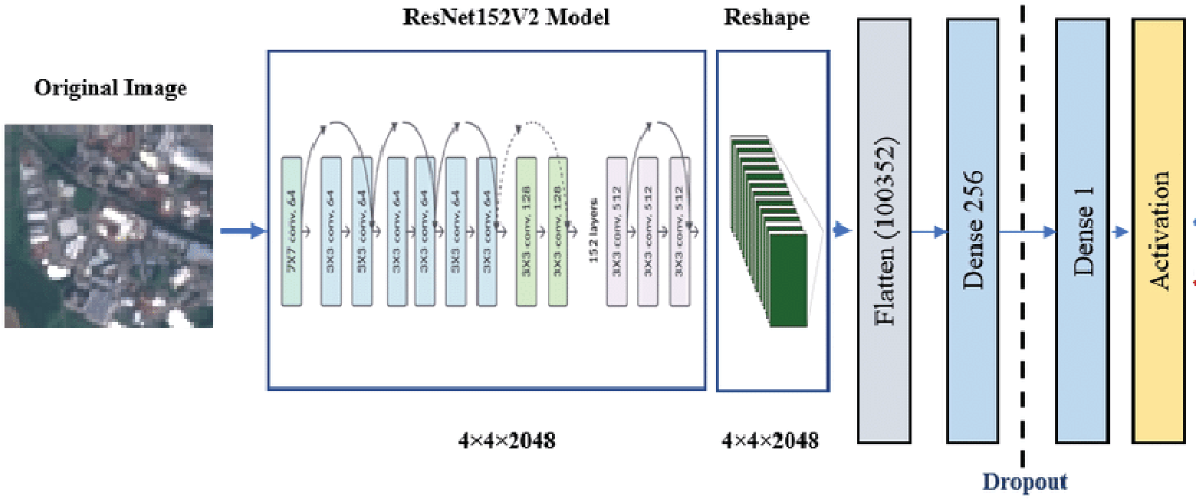


Figure 3.15: Architecture of ResNet152V2 (Source: Google)

e) Xception

A deep convolutional neural network The Xception, (Extreme Inception) model, which uses depthwise separable convolutions, implemented by separating the spatial and channel-wise processing into two steps, first by a depthwise convolution produced (on each channel separately) and then by a pointwise (1x1) convolution to recombine channel information. The design allows better efficiency in learning and minimizes the cost of computation. Xception is designed in three different flows, entry, middle and exit flows. Basic characteristics are extracted by the entry flow and spatial dimensions are also reduced. The middle flow is made of a block (repeated in identity) and provides a greater representation of features without dimensional transformations. Global average pooling and dense layer are applied to refine features in the exit flow in preparation to classification. At each node of the network, there exist residual connections to maintain stable flow of gradients and a quicker convergence. In complete separation of the spatial and cross-channel learning, Xception optimizes the representational power and performance. Figure 3.16 contains its architecture [58].

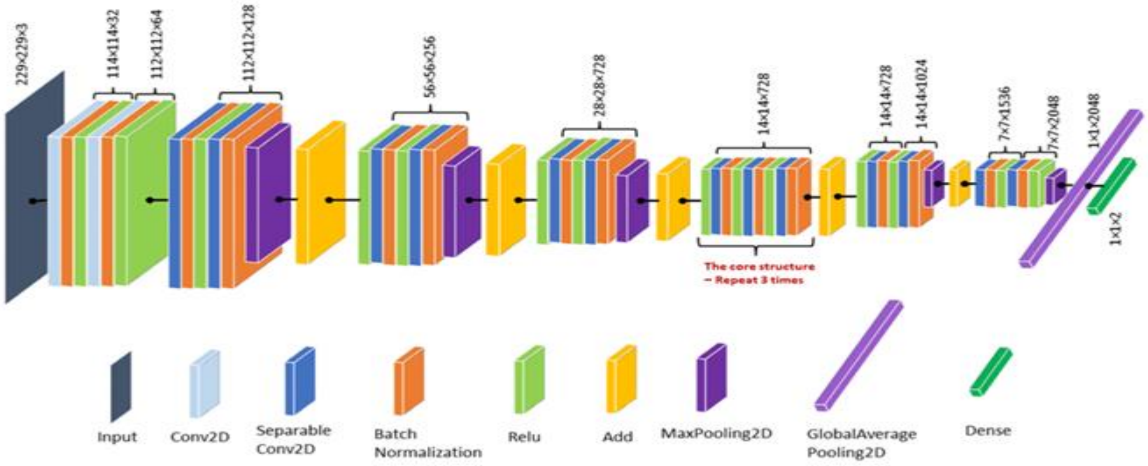


Figure 3.16: Architecture of Xception (source: Google)

f) ResNet101

The structure of ResNet101 is identical in terms of principles of architecture among other models in the ResNet family, yet it is less deep and has a lower computational cost. It has the use of residuals blocks which include shortcuts which make the network to train deeper and use more abstract features efficiently by enabling flow of a gradient.

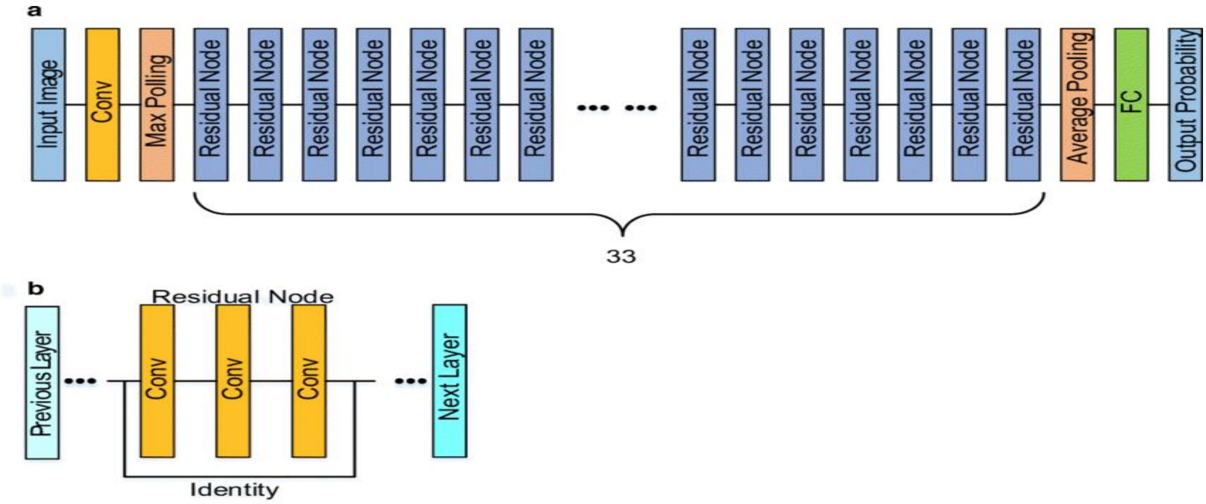


Figure 3.17: Architecture of ResNet101 (Source: Google)

This architecture proves useful when identifying patterns and textures in multifarious leaf images where symptoms of the disease could be of different shape, size, and intensity. Examples of such patterns would be intricate architectures, and because of the depth of the ResNet101, it is able to learn complex patterns, when compared to overfitting. ResNet101 is a stable benchmark model, which has been utilized many times in relation to image classification challenges, so it is possible to compare it with other variations in the current study, according to the proven performance results. The Figure 3.17 shows the structural construction of ResNet101 [59].

3.2.4 Proposed AgriViTX Model

The central aim was to suggest such a AgriViTX, a new Vision Transformer (ViT) structure that had the potential to fine tune several types of crop illness classification specifically. Global contextual sense relating to the computational efficiency is the notion that the driving force undertaken in the study of this proposed model development was balanced. In addition, it was also intended to scale the disadvantages of CNNs in modeling the long-range dependencies in keeping also in mind that it was possible to avoid the high computational costs that is commonly found with the standard transformer-based models.

Algorithm 1 ViT-Based Multi-Crop Disease Classification and Explainability

```
Input: Crop image dataset D
Parameters: $S$ = 128 (image size), $C$ = 3 (channels), $B$ = 32 (batch size), $P$ = 32 (patch
  size), $L$ = 6 (transformer layers), $H$ = 4 (attention heads), $D$ = 128 (hidden dim), $E$ = 100
  (epochs)
Output: Trained ViT model, evaluation metrics, visualizations, XAI
Procedure AgriViTX Workflow
// Data Loading and Preprocessing
        Count total images $N$ in D; assert $N > 0$
        Split D into training (70%), validation (20%), and test (10%) sets
        Resize all images to 128 × 128; normalize pixel values to [0,1]
        // Patch Extraction
        for each image $I$ do
        Divide $I$ into 4 × 4 = 16 non-overlapping patches of size 32 × 32
        Flatten each patch to a vector of length 32×32×3=3072
        // Model Construction
        Input: Tensor of shape (128,128,3)
        Apply feed-forward network: Dense(512, GELU) → Dense(128, GELU) →
  Dropout(0.1) → Residual
end for
        Extract class token output for prediction
        // Model Compilation and Training
        Compile with Adam optimizer (lr=0.0001), SparseCategoricalCrossentropy
Train for 100 epochs with early stopping, model checkpoint, and ReduceLROnPlateau
        // Model Evaluation
        Evaluate on test set: compute accuracy, loss
        // Explainability and Visualization
    Generate SHAP, LIME explanations, and Pseudo Segmentation
end procedure
```

To do this, the architecture incorporates optimized components of ViT, such as patch embedding, and the multitype self-attention. They were critically built with less complexity of parameters hence making the model fit to process high resolution crop images. Moreover, it added changes to the model so that it fits specificities of the domain of medical datasets, as it has to be both high-performance and interpretable.

Its development was emphasized in its architecture (Figure 3.18), reasoning, and mathematical implementation of the proposed AgriViTX model. Algorithm 1 of the complete workflow contains the key steps occurred in work, starting with initial ones and finished with final.

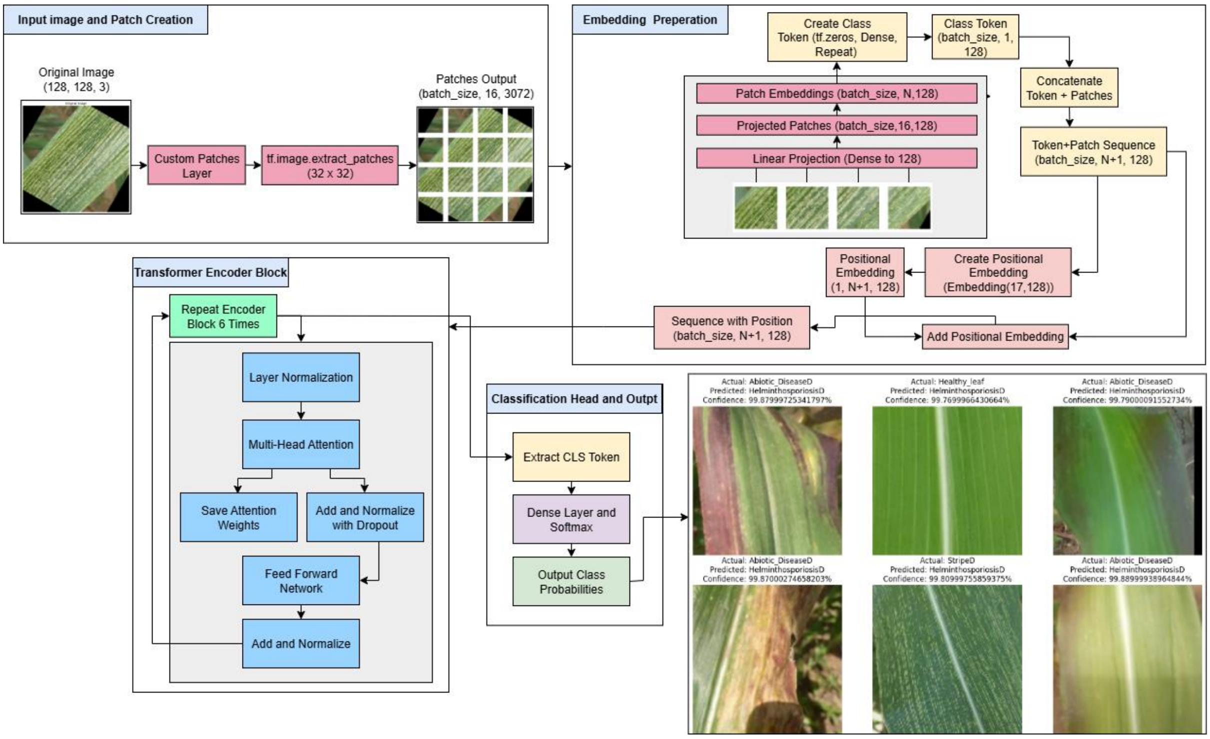


Figure 3.18: Proposed Model Architecture.

Table 3.2: Parameter summary of the proposed architecture

Parameter Type	Count	Total Parameters	Size
Trainable Parameters	2,772,503	2,772,503	10.58 MB
Non-trainable Parameters	0	0	0.00 MB
Total Parameters	**2,772,503**	**2,772,503**	**10.58 B**

3.2.6 Explainable AI Integration

a) SHAP

SHAP is a very strong interpretability tool that measures the impact of each feature on the outcome, such that a prediction can be decomposed into its component features. It is more specific with the application of game theory Shapley values to give a score of contribution to each input [60]. In this paper, it was used to not only identify the final decision made by the model but also the effort in each of the factors that led to the decision.

It assisted it to visualize which image areas influenced the model to assign it into one of the disease classes with red being a positive effect, and blue signifying a negative one. This provided both international and local context of model-behavior that helps in advancing the decision-making process to a very candid and reliable process. An equation of the SHAP is obtained as below [61].
For a model f, input image x, and a baseline x', the SHAP value for feature (pixel/region) i is:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} [f(S \cup \{i\}) - f(S)] \qquad (1)$$

b) LIME

LIME (Local Interpretable Model-Agnostic Explanations) constructs local linear models by fixing around certain predictions to estimate the behaviour of black-box models [62]. On image data, it will divide a picture into superpixels and indicate the parts that are superpixels most affected the model. LIME was adopted to zoom in on single forecasts such as the areas of the leaves that contributed to the model classifying a sample as either an early blight or leaf mold. It was particularly useful in demonstrating model behaviour case by case which is vital when both debugging and explanation is in order. LIME interprets the predictions of any classifier at a local level by locally approximating the model in terms of an interpretable model (e.g. linear regression or decision trees). The general derivation of the equation is the following [63].

The input images are segmented into k superpixels. A binary vector $z' \in \{0,1\}^d$ is used to indicate which superpixels are present in a perturbed version of the original image. LIME learns a linear model $g(z')$, where:

$$g(z') = w \cdot z' = \sum_{j=1}^{k} w_j z'_j \qquad (2)$$

The local loss gets minimized with complexity control. To find the best explanation model $\boldsymbol{g}$, LIME minimizes a loss function that balances approximation quality and simplicity:

$$\arg \min_{g} L(f, g, \pi_x) + \Omega(g) \qquad (3)$$

c) Pseudo Segmentation

Pseudo segmentation is a semi-supervised or unsupervised method that in approximation seeks to model pixel-level (or region-level) segmentation maps using image-level labels or model attention [64]. Pseudo segmentation produces rough spatial maps of where there may be something of interest or importance of a particular class prediction, rather than demanding completely annotated masks. Pseudo segmentation can be computed, in the case of vision transformers (or other classification models), by retrieving and processing the output of attention heatmaps or class activation maps (among others), to identify areas of greatest relevance to the model decision [65]. These rough masks can be used to understand which regions of an image are used to classify the image without manual pixel-wise labeling. Pseudo segmentation is therefore seen as an intermediate between classification and segmentation analysis and provides useful visual interpretation and allows downstream interpretation such as disease localization or regional analysis.

3.3 Project Plan

This study has a well-designed project plan that organizes the workflow in a logical goal-oriented pattern with the objective of creating a reliable, interpretable, and deployable system that can aid in crop leaf disease detection. The study begins with data gathering where a complete image set of the leaves of various crops infected by various diseases is gathered together with different sources. During data preprocessing the size of images is rescaled, they are all normalized and the labels are encoded in order to prepare the data to be fitted. To resolve the problem of the class imbalance that usually exists in agricultural datasets, a number of data augmentation methods are used that are applied solely to the training set to provide the positive outcome of the equality of the representation of all classes of the disease. Multiple architectures of the convolutional neural network (CNN) are then trained and tested in order to determine the baseline measurements in crop-disease classification. On this note, the concept of transfer learning has been introduced where there is the freezing of base layers of these models and new training of the classification heads in order to attain better accuracy and short training time. The key contribution of the work is that it is the design and training of custom Vision Transformer (ViT) architecture as the basis of crop leaf disease detection. Trained by scratch, this ViT model is optimized to represent complex spatial dependencies and subtler patterns of disease in comparison to traditional CNNs. The most successful ViT model also uses explainable AI (XAI) methods, like SHAP, LIME, and attention-based visualization to enable explainability of the model classification decisions, improving the transparency and interpretability of the model. In such a way, this method will help not only to come up with the system that has good performance but also to generate credible explanations that are critical towards applications in agriculture.

3.4 Task Allocation

The project of this study will be executed according to a systematic and objective workflow and it will go through several major phases to come up with an accurate, explainable, and deployable disease detection system of crops (Maize, Tomato, Onion). The research will start with the stage of data collection when a complete set of most of the crop images can be retrieved through an online repository. After that, the second step is resizing, normalization, and label encoding in the data preprocessing stage to represent the images to be used in the modeling. Class imbalance can also be alleviated by data augmentation where rotation, flipping, scaling, etc. are performed on the training data only so that balanced learning occurs among all disease classes. Then various established fine-tuned transfer learning models are used wherein the base layers of the models are kept frozen and the top classification layers are retrained to make the performance better. The performance of the results is extended where a new and custom Vision Transformer (ViT) model is proposed and trained on scratch to utilize its better ability in capturing complex spatial relations in crop leaf image.

The research aim is to determine the possibility of ViT model to obtain better accuracy and gain the benefits of explainability compared to the conventional CNN models. To make the model transparent and predictable, the model that achieved the highest performance is combined with Explainable AI (XAI) methods, SHAP, LIME, attention-based visualization with pseudo segmentation methods.

Table 3.3: Task Allocation

Tasks	**Weeks**														
	12	14	16	18	20	22	24	26	28	30	32	31	32	21	32
Working Plan	🟦	🟦													
	🟩	🟩													
Theoretical Study		🟦	🟦												
		🟩	🟩												
Literature Review		🟦	🟦												
		🟩	🟩												
Data Collection and Preparation			🟦	🟦											
			🟩	🟩											
Transfer Learning Experiment					🟦	🟦	🟦								
					🟩	🟩	🟩	🟩							
Proposed Model Development								🟦	🟦						
								🟩	🟩	🟩					
Web App Development										🟦	🟦	🟦			
										🟩	🟩	🟩	🟩		
Report Writing											🟦	🟦	🟦		
											🟩	🟩	🟩		
Review and Finalization													🟦	🟦	🟦
													🟩	🟩	🟩

3.5 Summary

This chapter described the methodology followed in the development of explainable and optimized deep learning-based framework to detect the disease in the leaves of maize, tomato, and onion. The strategy involved the process of getting a public dataset (TOM2024) of high quality and thoroughness, the subsequent preprocessing, normalization, and augmentation of the dataset with the application of particular methods with both PIL and OpenCV mathematical libraries to increase the diversity and balance of the dataset. The experimental setup was represented by testing six well-developed transfer learning models and a self-built Vision Transformer (ViT) model that can be trained on top. A wide range of experiments and ablation studies were performed to optimize the hyperparameters and test how generalizable the proposed model would be in regard to many different crops. To enhance the comprehensibility and interpretability, explainable AI (XAI) methods, including LIME, SHAP, and attention-based heatmaps, were implemented so one can have a visualization of the choices the model is making. In order to create areas of interpretation better, pseudo-segmentation was also performed to mark out areas with disease presence. The design elements that were at system level were presented in the form of context diagrams, level 1 data flow diagrams (DFD) and user interface mock ups to achieve usability and functional clarity. Separation of user interaction and backend computation was done so that the underlying computations occur in the backend, thereby creating a smooth working process with modularity, achieved through a two-tier architecture. Besides, there was also an elaborated project plan, which would help to proceed through the project scrupulously stage by stage, starting with data preparation and moving up to model deployment.

Chapter 4

Implementation and Results

This chapter shows the results of the analysis of the performance of the developed deep learning framework applied to the design of a multi-crop leaves disease classification system, which, in particular, concerns maize, tomato, and onion. The first part summarizes the entire experimental framework, including the description of the hardware and software environments of the model training and testing processes. Subsequently, the chapter includes the results of several optimized Convolutional Neural Network (CNN) and the Vision Transformer (ViT)-based transfer learning models used on the balanced dataset. Data augmentation methods were also used to achieve a balanced parity in terms of the distribution of classes within the dataset that was made robust by increasing diversity. Further, a CNN-ViT hybrid architecture constructed to enhance classification accuracy is presented in the chapter. Lastly, a range of Explainable AI (XAI) techniques are used to explain model predictions and increase transparency, to better understand how the model assigns class labels to disease-affected leaves across crops.

4.1 Environment Setup

This research study was carried out with the help of Python 3.10, with TensorFlow and Keras at the foundation of the deep learning models. These processes of training and testing were carried out on two main computing mediums to enhance flexibility and effectiveness, Kaggle Notebooks and a personal computer. Kaggle Notebooks offered a cloud-hosted system with NVIDIA Tesla P100 and T4 GPUs (allocated on-demand), 13GB RAM, and about 20GB of temporary storage space, and have up to 30 hours of free computing weight-time each week. This was used to support faster model training and experimentation on strong hardware decoupled from local resource availability.

Table 4.1 summarizes the key software, libraries, and hardware components used throughout the study.

Component	Description
Platforms	Kaggle Notebooks, Personal PC
Kaggle GPU	NVIDIA Tesla P100 / T4 (Dynamic Allocation)
Kaggle RAM	13 GB
Kaggle Disk	~20 GB Temporary
Personal PC CPU	AMD Ryzen 7 5600H
Personal PC GPU	NVIDIA GeForce RTX 3060
Personal PC RAM	16 GB
Python Version	3.1
Key Libraries	TensorFlow, Keras, NumPy, Pandas, OpenCV, scikit-learn, Matplotlib, imbalanced-learn, SHAP, LIME
Deployment Tools	Fast API (Web)

The personal computing platform consisted of a Ryzen 7 5600H processor, an NVIDIA GeForce RTX 3060 GPU, and 16 GB of RAM, the powerful local environment to support model development, debugging, and testing. This allowed large-scale computation-heavy programs to be handled easily and remain flexible to development. Key libraries like NumPy, Pandas, Matplotlib, OpenCV and scikit-learn were incorporated in the framework to work with data, and visualize. Besides, imbalanced-learn was adopted in performing to manage the imbalance of the classes during training. The end result was the comparison of the models to determine the best performing model and what was found was selected to undergo an interpretability analysis using LIME and SHAP techniques. The model was then converted to formats supporting TensorFlow Lite as well as .h5 files which were used to deploy the model. The framework has been established with regard to set world standards in engineering to be compatible, adaptable, suitable privacy of data, and usability of software in different applications. Table 4.1 presents a description of the main programs and libraries used during the research.

Table 4.2 Model Training Parameters

Parameter	Value
Optimizer	Adam
Learning Rate	0.0001
Loss Function	Sparse Categorical Crossentropy (from_logits=False)
Metrics	Accuracy
Epochs	Up to 100 (with EarlyStopping)
Batch Size	32
Callbacks	ModelCheckpoint (monitor: val_accuracy, save best only, save weights only)
	EarlyStopping (monitor: val_loss, patience: 15, restore best weights)
	ReduceLROnPlateau (monitor: val_loss, factor: 0.5, patience: 10, cooldown: 2, min_lr: 1e-6)

The information obtained in the regional plantations was pre-processed where all the images were resized to a common size of 224 x 224 pixels. Next, the data augmentation and normalization techniques were used to improve the quality of dataset. The data was also subdivided into training, validation and testing sets with percentages of 70, 10 and 20 respectively. Also, the EarlyStopping, ReduceLROnPlateau, and ModelCheckpoint features were added to enhance the performance of the model training and avoid overfitting. These callbacks were designed to monitor the performance metrics as well as to adaptively adjust parameters. Training would stop or be halted based on trends in validation loss.

4.2 Testing and Evaluation

When testing the accuracy of models trained to classify leaf diseases, a number of key metrics have been deployed to make the assessment thorough and credible. Not only are such metrics significant to understanding the overall accuracy of the models but also their robustness, error type and generalizability which are critical when considering plant disease classification application where misclassifications have serious implications on agricultural practice.

4.2.1 Precision

Precision (as presented in Equation 12) was used to calculate the number of the samples that were correctly predicted as a particular disease category. It has been quantified as the true positives divided by the total of the true positives and false positives. The greater the precision score, the more likely that when the model predicted a positive case, it was quite correct and did not create excessive false alarms.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (4)$$

4.2.2 Recall

The measure of the number of actual disease-positive cases that were fully detected by the model was determined by the Recall (Equation 13). Since this study is more medical-oriented, it was important to obtain a higher recall value. It ensures that no cases with diseases went without checks.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (5)$$

4.2.3 F1-Score

F1-Score (Equation 14) may be viewed as a combination of precision and recall. It was used especially because in certain situations there can be a trade-off between the number of true positive outcomes (recall) and the accuracy of such predictions (precision). That is why the harmonic mean of these metrics was also calculated to demonstrate the entire picture.

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (6)$$

4.2.4 Accuracy

Accuracy (Equation 15) calculated the number of all predictions made by the model that were accurate. It considered correct negatives and correct positives. Nevertheless, it was held in check by other metrics because class imbalance of medical data could cause inaccurate accuracy in some cases.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad (7)$$

4.2.5 Specificity

It was also added specificity (Equation 16) or the true negative rate, to ensure that the model was not missing the false positives to be precise how many of the healthy or non-targets were correctly discarded by the model. A greater specificity ensured that the model did not incorrectly conclude that a large number of samples that are observed to be healthy are samples that have been developed with the disease.

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

4.3 Results and Discussion

This section provides the various results and main findings of experimental evaluations carried out in this research. A thorough set of classification metrics is used to evaluate the performance of six systematically optimized transfer learning CNN models ResNet152V2, Xception, InceptionV3, Inception ResNet V2, and DenseNet201, ResNet101, and the proposed custom Vision Transformer (ViT) model. Afterwards, the top-performing ViT model is subjected to Explainable AI (XAI) methods to increase the transparency and interpretability of the predictions. The strength and applicability of the proposed framework is also demonstrated by its implementation as a web-based interactive application prototype.

4.3.1 Results on The Tomato Dataset

a) Classification Report of ViT

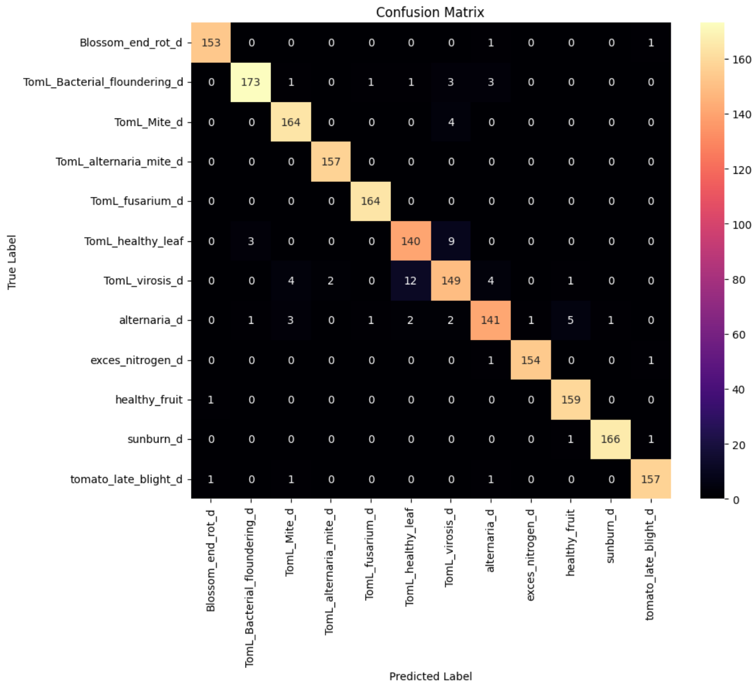
Table 4.3 shows the classification results on the Tomato Dataset of the proposed Vision Transformer (ViT) model. The findings indicate that the model performs well on all classes with an overall accuracy of 96%. The majority of the disease categories, including Blossom_end_rot_d, TomL_fusarium_d, sunburn_d, and exces_nitrogen_d received extremely high precision, recall, and F1-scores (approaching 0.99), which demonstrates the strength of the model. But, relatively lower scores were obtained on TomL_virosis_d and TomL_healthy_leaf, which implies that these classes were a bit more difficult to predict by the model. In general, the ViT model shows a stable and effective classification ability to identify tomato diseases.

Table 4.3: Classification report of the proposed ViT model on the Tomato Dataset

Class	Precision	Recall	F1-Score	Support	Accuracy
Blossom_end_rot_d	0.99	0.99	0.99	155	**0.98**
TomL_Bacterial_floundering_d	0.98	0.98	0.96	182	
TomL_Mite_d	0.99	0.98	0.96	168	
TomL_alternaria_mite_d	1.00	1.00	0.99	157	
TomL_fusarium_d	0.99	0.99	0.99	164	
TomL_healthy_leaf	0.99	0.98	0.95	162	
TomL_virosis_d	0.97	0.97	0.98	172	
alternaria_d	0.98	0.98	0.98	157	
exces_nitrogen_d	0.99	0.99	0.99	156	
healthy_fruit	0.99	0.98	0.98	160	
sunburn_d	0.99	0.99	0.99	168	
tomato_late_blight_d	0.98	0.98	0.98	160	
Macro Avg	0.98	0.98	0.98	1951	
Weighted Avg	0.98	0.98	0.98	1951	

b) Confusion Matrix of ViT

In order to better assess the classification performance of the model, a confusion matrix was also created in Figure 4.1, which gives more details concerning the misclassifications of the classes. The confusion matrix has proved that the majority of errors were produced within the visually similar groups, like TomL_virosis_d and TomL_healthy_leaf, whereas other classes of diseases were not overlapped significantly. This additional discussion enhances the validity of the ViT model in the correct diagnosis of tomato leaf and fruit diseases.



True Label \ Predicted Label	Blossom_end_rot_d	TomL_Bacterial_floundering_d	TomL_Mite_d	TomL_alternaria_mite_d	TomL_fusarium_d	TomL_healthy_leaf	TomL_virosis_d	alternaria_d	exces_nitrogen_d	healthy_fruit	sunburn_d	tomato_late_blight_d
Blossom_end_rot_d	153	0	0	0	0	0	0	1	0	0	0	1
TomL_Bacterial_floundering_d	0	173	1	0	1	1	3	3	0	0	0	0
TomL_Mite_d	0	0	164	0	0	0	4	0	0	0	0	0
TomL_alternaria_mite_d	0	0	0	157	0	0	0	0	0	0	0	0
TomL_fusarium_d	0	0	0	0	164	0	0	0	0	0	0	0
TomL_healthy_leaf	0	3	0	0	0	140	9	0	0	0	0	0
TomL_virosis_d	0	0	4	2	0	12	149	4	0	1	0	0
alternaria_d	0	1	3	0	1	2	2	141	1	5	1	0
exces_nitrogen_d	0	0	0	0	0	0	0	1	154	0	0	1
healthy_fruit	1	0	0	0	0	0	0	0	0	159	0	0
sunburn_d	0	0	0	0	0	0	0	0	0	1	166	1
tomato_late_blight_d	1	0	1	0	0	0	0	1	0	0	0	157

Figure 4.1: Confusion Matrix of Tomato

c) Plot Training History of ViT

Figure 4.2 shows the history of training and validation of the proposed Vision Transformer (ViT) model. The accuracy curve is monotonically increasing with epochs and the model reached accuracy of 98%, the curve has stabilized showing that the model had converged. In a similar way, the loss graph indicates an overall decrease in training and validation, which proves that the model was efficient in reducing the number of errors in classifications. The fact that training and validation curves were very close to each other also implies that no overfitting happened in the model and that it generalized well.

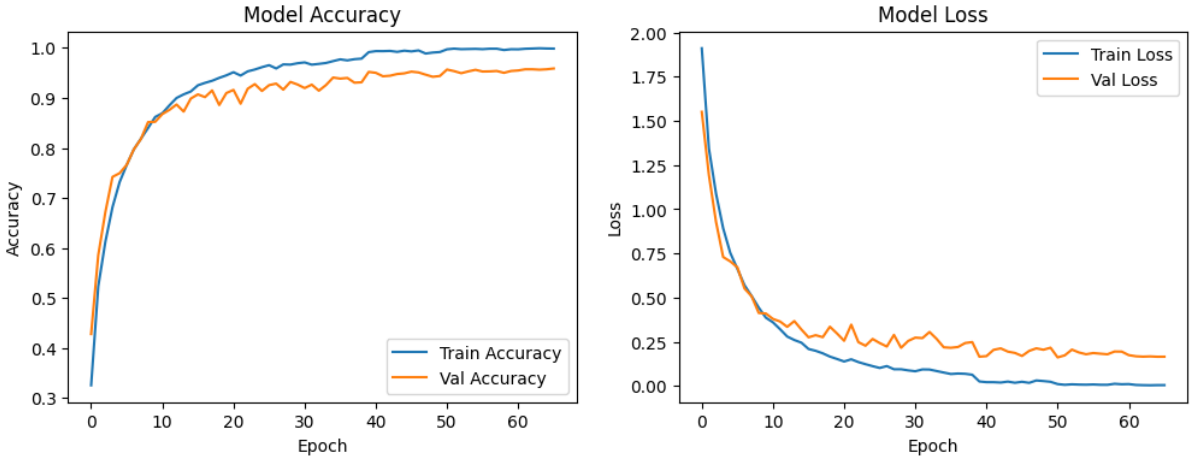


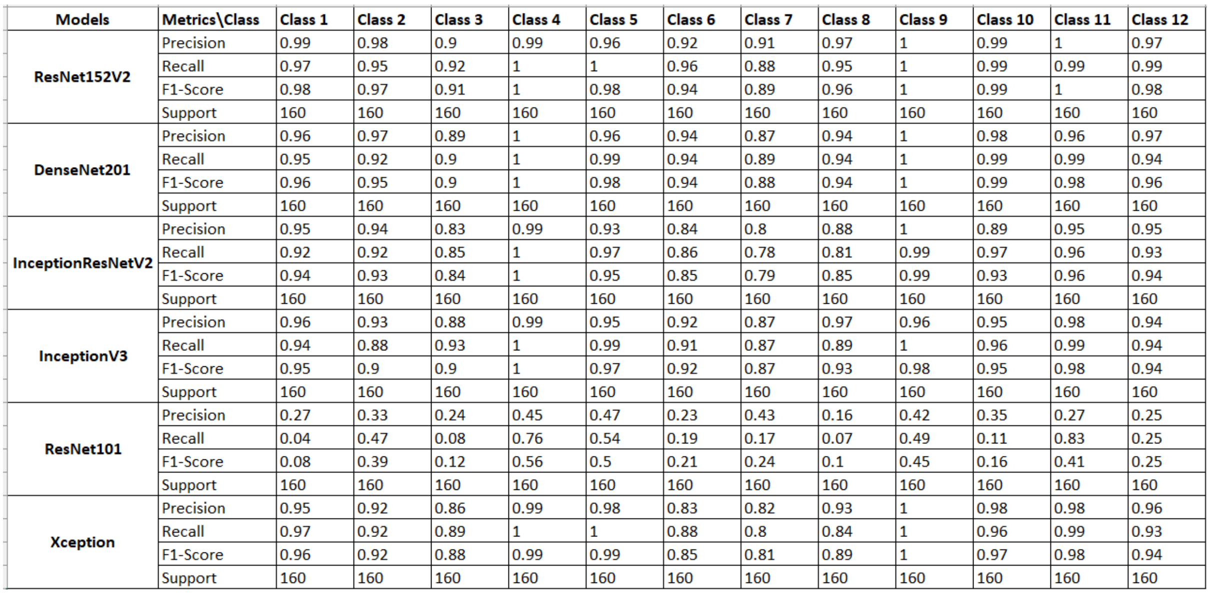
Figure 4.2: Plot Training of Tomato

d) Comparison with Baseline CNN Models

To test the efficiency of the suggested Vision Transformer (ViT) model, its work was compared with some of the baseline models. The comparison was done with help of the classification reports and training history plots (accuracy and loss curves) of each model.

The ViT classification report, presented in Table 4.4, shows that it obtained an overall accuracy of 98%, beating most of the baseline methods. Specifically, the ViT model demonstrated nearly perfect precision and recall on such categories as Blossom end rot d, TomL fusarium d, and sunburn d, and lower but still strong results on TomL virosis d and TomL healthy leaf. This shows that ViT model is more efficient in the differentiation of complex patterns of diseases than the conventional architectures.

Table 4.4: Classification Report of the Baseline CNN Models



Models	**Metrics\Class**	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class 5**	**Class 6**	**Class 7**	**Class 8**	**Class 9**	**Class 10**	**Class 11**	**Class 12**
ResNet152V2	Precision	0.99	0.98	0.9	0.99	0.96	0.92	0.91	0.97	1	0.99	1	0.97
	Recall	0.97	0.95	0.92	1	1	0.96	0.88	0.95	1	0.99	0.99	0.99
	F1-Score	0.98	0.97	0.91	1	0.98	0.94	0.89	0.96	1	0.99	1	0.98
	Support	160	160	160	160	160	160	160	160	160	160	160	160
DenseNet201	Precision	0.96	0.97	0.89	1	0.96	0.94	0.87	0.94	1	0.98	0.96	0.97
	Recall	0.95	0.92	0.9	1	0.99	0.94	0.89	0.94	1	0.99	0.99	0.94
	F1-Score	0.96	0.95	0.9	1	0.98	0.94	0.88	0.94	1	0.99	0.98	0.96
	Support	160	160	160	160	160	160	160	160	160	160	160	160
InceptionResNetV2	Precision	0.95	0.94	0.83	0.99	0.93	0.84	0.8	0.88	1	0.89	0.95	0.95
	Recall	0.92	0.92	0.85	1	0.97	0.86	0.78	0.81	0.99	0.97	0.96	0.93
	F1-Score	0.94	0.93	0.84	1	0.95	0.85	0.79	0.85	0.99	0.93	0.96	0.94
	Support	160	160	160	160	160	160	160	160	160	160	160	160
InceptionV3	Precision	0.96	0.93	0.88	0.99	0.95	0.92	0.87	0.97	0.96	0.95	0.98	0.94
	Recall	0.94	0.88	0.93	1	0.99	0.91	0.87	0.89	1	0.96	0.99	0.94
	F1-Score	0.95	0.9	0.9	1	0.97	0.92	0.87	0.93	0.98	0.95	0.98	0.94
	Support	160	160	160	160	160	160	160	160	160	160	160	160
ResNet101	Precision	0.27	0.33	0.24	0.45	0.47	0.23	0.43	0.16	0.42	0.35	0.27	0.25
	Recall	0.04	0.47	0.08	0.76	0.54	0.19	0.17	0.07	0.49	0.11	0.83	0.25
	F1-Score	0.08	0.39	0.12	0.56	0.5	0.21	0.24	0.1	0.45	0.16	0.41	0.25
	Support	160	160	160	160	160	160	160	160	160	160	160	160
Xception	Precision	0.95	0.92	0.86	0.99	0.98	0.83	0.82	0.93	1	0.98	0.98	0.96
	Recall	0.97	0.92	0.89	1	1	0.88	0.8	0.84	1	0.96	0.99	0.93
	F1-Score	0.96	0.92	0.88	0.99	0.99	0.85	0.81	0.89	1	0.97	0.98	0.94
	Support	160	160	160	160	160	160	160	160	160	160	160	160

Moreover, the training and validation history plots introduced in Figure 4.5 show that the ViT model had converged with stable learning patterns. The training and validation curves of the accuracy were very close and the loss kept on decreasing with no major variation which indicated less overfitting in comparison to the baseline models.

Table 4.5: Classification Report of CNN Models

Model	Train Accuracy	Validation Accuracy	Test Accuracy	Train Loss	Validation Loss	Test Loss
ResNet152V2	0.9865	0.9628	0.9656	0.0783	0.1405	0.1166
DenseNet201	0.9725	0.9498	0.9542	0.1282	0.1746	0.1597
InceptionResNetV2	0.9436	0.9289	0.9135	0.2212	0.2577	0.2740
InceptionV3	0.9705	0.9378	0.9406	0.1361	0.2070	0.1871
ResNet101	0.3398	0.3359	0.3349	1.9697	1.9849	1.9716
Xception	0.9653	0.9320	0.9323	0.1622	0.2238	0.2230

ResNet152V2 performed best overall overall with the highest test accuracy (96.6%) and the lowest test loss (0.1166) followed by DenseNet201. InceptionResNetV2 and InceptionV3 performed moderately with an accuracy of 91-94 percent. Xception was comparable to InceptionV3, whereas ResNet101 performed very poorly with an accuracy of ~33% and a very high loss.

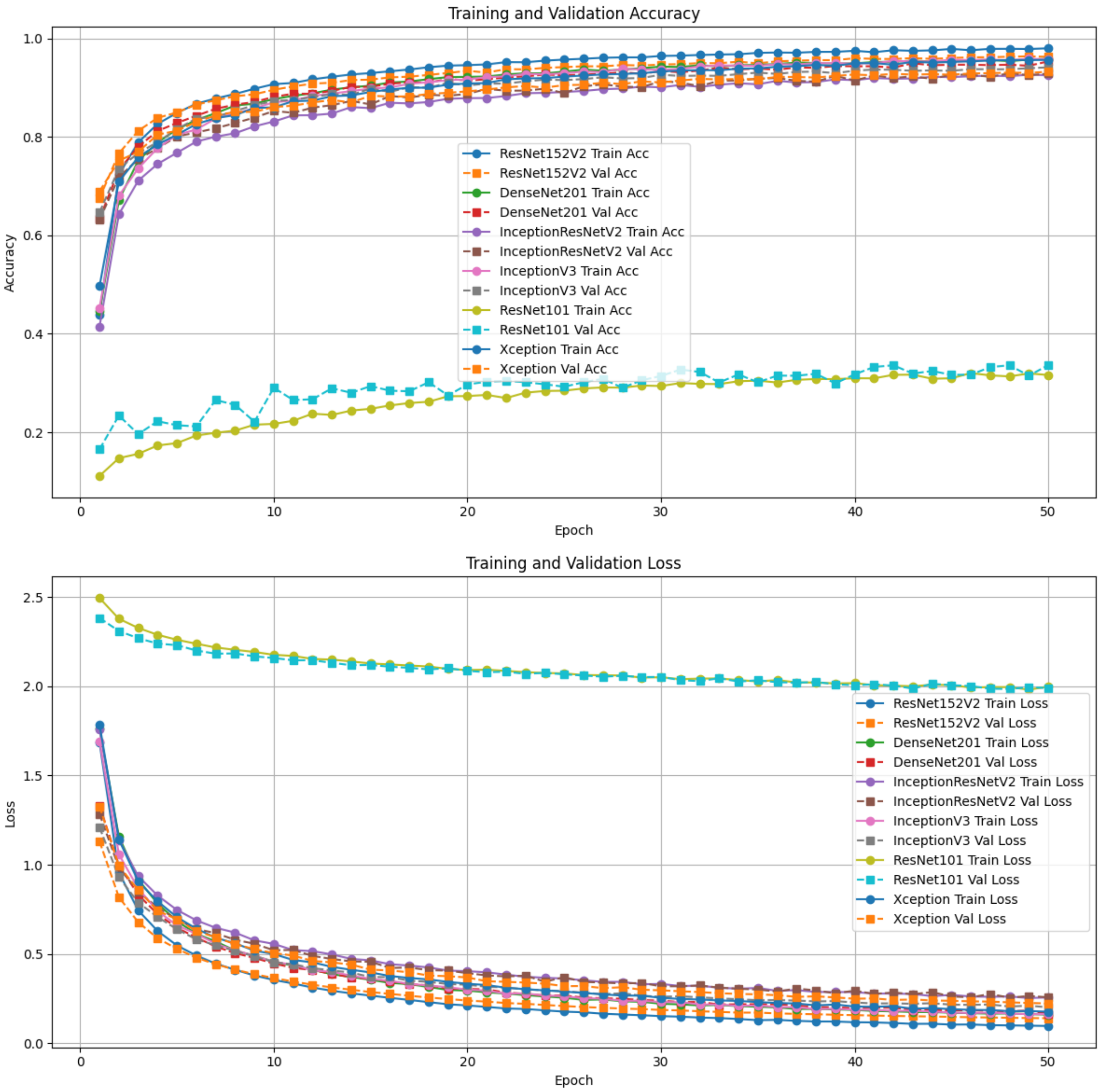


Figure 4.3: Training and validation curves (loss and accuracy) for each fine-tuned transfer learning models for Tomato.

Generally, the classification metrics and training history curves show that the proposed ViT model yields better classification results and generalization ability compared to the baseline CNN-based models.

4.3.2 Results on Onion Dataset

a) Classification Report

Table 4.6 shows the classification performance of the proposed Vision Transformer (ViT) approach on this dataset. The model obtained a total accuracy of 98%, which shows its good potential to separate the different disease classes and healthy leaves. The precision, recall, and F1-scores of the most classes were very high (approximately 0.98), indicating that the model was highly appropriate in detection of diseased and healthy samples.

Table 4.6: Classification Report of ViT

Class	Precision	Recall	F1-Score	Support	Accuracy
Alternaria_D	0.97	0.97	0.97	80	**0.96**
Fusarium_D	0.96	0.96	0.96	80	
Healthy_leaf	0.98	0.98	0.98	63	
Virosis_D	0.96	0.96	0.96	73	
Macro Avg	0.96	0.96	0.96	296	
Weighted Avg	0.96	0.96	0.96	296	

A slight decrease was also noted when it comes to Alternaria_D, with F1-score 0.97, indicating that this class was a little more difficult to classify, potentially because of its visual similarity with other types of diseases. Generally, the outcomes show the ViT model delivered strong and high-performance classification, which can be used to detect and control crop diseases.

b) Confusion Matrix

Along with the classification report, a confusion matrix was created to get a closer look at the performance of the ViT model on a per-class basis, as demonstrated in Figure 4.4.

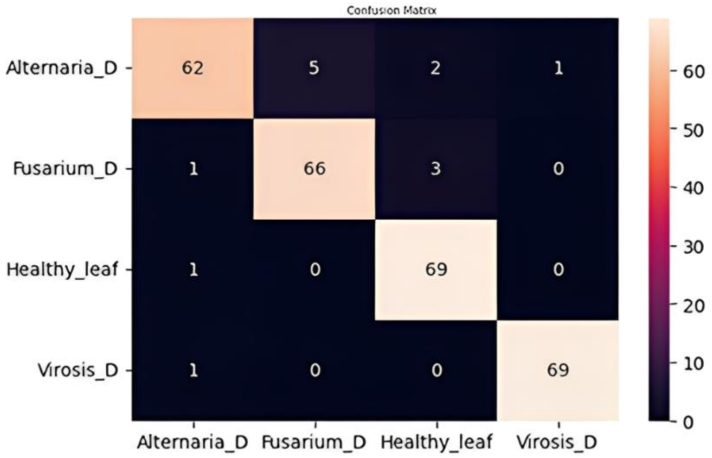


Figure 4.4: Confusion Matrix Onion

The matrix is highly dominant diagonally meaning that most samples were classified properly. There were minor misclassifications with the most occurring in Alternaria_D as there was some overlap with other classes of disease due to the visual similarity in patterns. In general, the confusion matrix demonstrates the strong verification and effectiveness of the ViT model regarding all classes.

c) Plot Training History

The training and validation history of the ViT model is depicted in Figure 4.5. The accuracy curves show a steady increase, stabilizing around 96%, while the loss curves demonstrate a consistent decline during training. The close alignment between training and validation curves suggests minimal overfitting, indicating that the model generalized well to unseen data. Together with the confusion matrix, these plots provide a comprehensive view of the model’s strong performance and effective learning behavior.

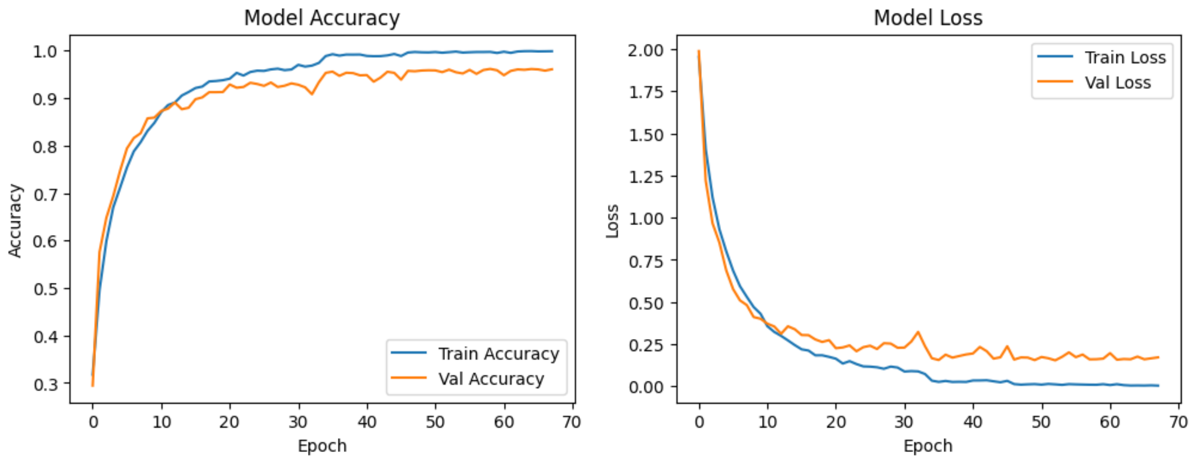


Figure 4.5: Plot Training History Onion

d) Comparison with Baseline CNN Models

The results of the classification accuracy of the six models used to determine the efficiency of each of the architectures in the identification of four categories of onion leaves show different trends. ResNet152V2 exhibited an outstanding overall performance with high precision and recall rates in all classes. It demonstrated the best F1-score of 0.99 on Virosis Disease and 0.96 on Healthy Leaf, resulting in an excellent multi-class classification system. DenseNet201 also performed well and consistently, with an overall excellent classification across all classes.

Table 4.7: Classification report of Onion Dataset

Model	**Class**	**Precision**	**Recall**	**F1-Score**	**Support**	**Accuracy**
ResNet152V2	*Alternaria_D*	0.95	0.89	0.92	70	0.95
	Fusarium_D	0.93	0.94	0.94	70	
	Healthy_leaf	0.93	0.99	0.96	70	
	Virosis_D	0.99	0.99	0.99	70	
DenseNet201	*Alternaria_D*	0.94	0.91	0.93	70	0.94
	Fusarium_D	0.97	0.93	0.95	70	
	Healthy_leaf	0.93	0.99	0.96	70	
	Virosis_D	0.97	0.99	0.98	70	
InceptionResNetV2	*Alternaria_D*	0.85	0.89	0.87	70	0.91
	Fusarium_D	0.92	0.86	0.89	70	
	Healthy_leaf	0.89	0.96	0.92	70	
	Virosis_D	0.99	0.94	0.96	70	
InceptionV3	*Alternaria_D*	0.90	0.89	0.89	70	0.93
	Fusarium_D	0.94	0.93	0.94	70	
	Healthy_leaf	0.93	0.99	0.96	70	
	Virosis_D	0.97	0.94	0.96	70	
ResNet101	*Alternaria_D*	0.40	0.20	0.27	70	0.33
	Fusarium_D	0.55	0.24	0.34	70	
	Healthy_leaf	0.42	0.70	0.53	70	
	Virosis_D	0.49	0.69	0.57	70	
Xception	*Alternaria_D*	0.89	0.83	0.86	70	0.92
	Fusarium_D	0.90	0.91	0.91	70	
	Healthy_leaf	0.92	0.97	0.94	70	
	Virosis_D	0.99	0.99	0.99	70	

The overall classification performance of the six models put to test shows clear patterns of the extent to which the various architectural models detected the four onion leaf types, Alternaria Disease, Fusarium Disease, Healthy Leaf, and Virosis Disease. ResNet152V2 showed a good overall performance and high precision and recall of all classes. Most significantly, it recorded an excellent F1-score of 0.99 in Virosis Disease and 0.96 in Healthy Leaf, which is a significant contribution to a very reliable multi-class detection system. The precision-recall tradeoff with balanced performance across categories demonstrates that it is robust regarding both healthy and diseased samples. DenseNet201 demonstrated similar and robust performance, and overall high classification of all classes.

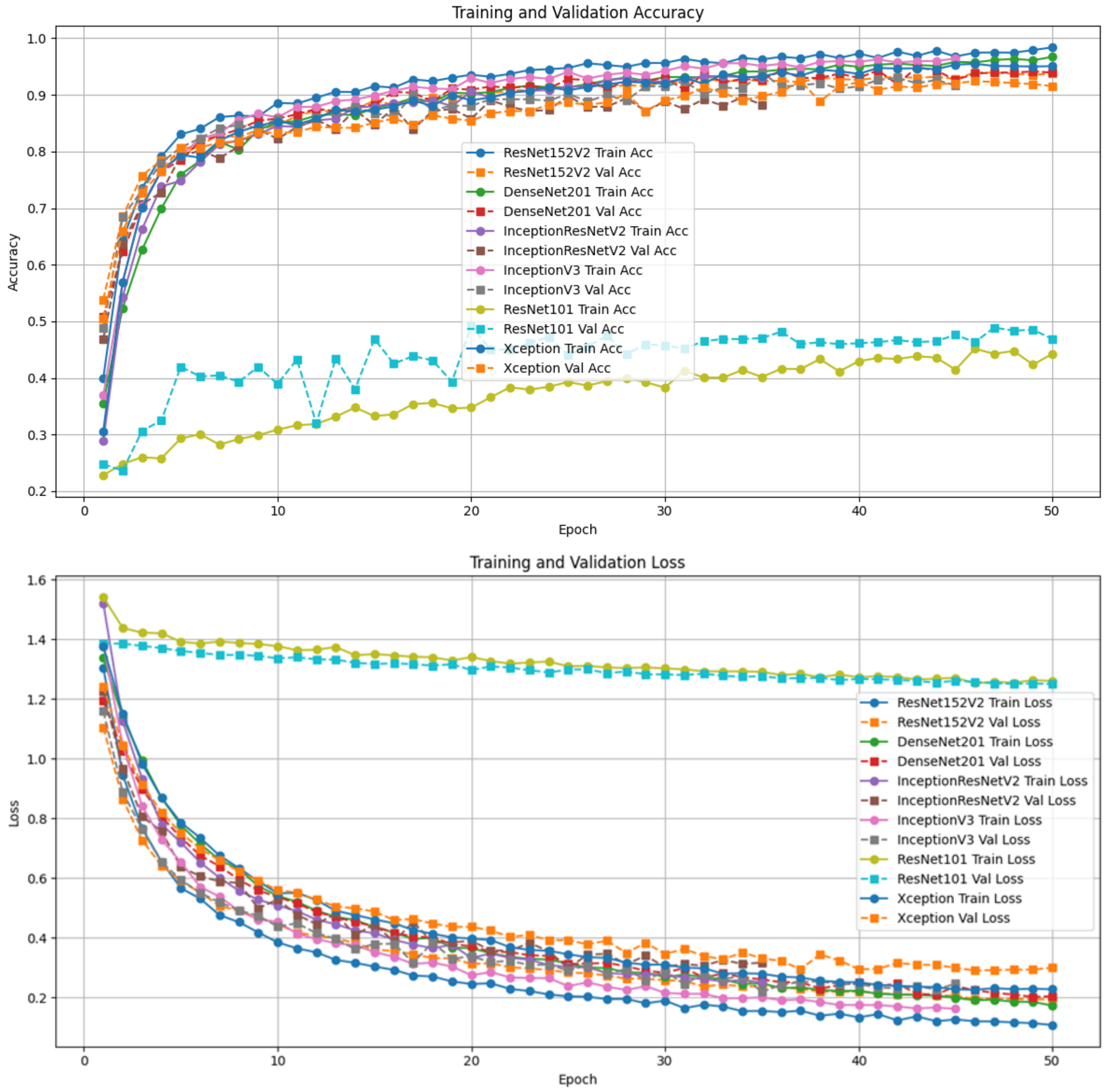


Figure 4.6: Training and validation curves (loss and accuracy) for each fine-tuned transfer learning model for Onion.

4.3.3 Results on Maize Dataset

a) Classification Report of ViT

Table 4.8 is the classification performance of the proposed Vision Transformer (ViT) model on the Maize Dataset. The model had an overall accuracy of 98 percent thus showing that it has a good discriminative power against the different maize leaf diseases. The high precision, recall, and F1-scores (≈0.99) in the majority of the classes (Abiotic_DiseaseD, CurvulariaD, and RustD) show that the model could correctly identify these diseases. Likewise, HelminthosporiosisD also performed very well with F1-score of 0.98. Conversely, Healthy_leaf, StripeD and VirosisD showed a performance that was slightly lower with F1-scores averaging 0.97. This indicates that these classes were a source of a relatively higher classification difficulty perhaps because they were visually similar to other classes of diseases. On the whole, the findings indicate that the ViT model is an excellent classification tool that has high accuracy and robustness when used in the detection of maize diseases hence an effective measure of aiding in disease management in maize crops.

Table 4.8: Classification Report of The Proposed ViT Model on The Maize Dataset

Class	Precision	Recall	F1-Score	Support	Accuracy
Abiotic_DiseaseD	1.00	1.00	0.99	230	**0.97**
CurvulariaD	0.99	0.98	0.99	245	
Healthy_leaf	0.97	0.97	0.97	211	
HelminthosporiosisD	0.97	0.99	0.98	225	
RustD	0.99	0.99	0.99	211	
StripeD	0.98	0.97	0.97	227	
VirosisD	0.97	0.97	0.97	199	
Macro Avg	0.97	0.97	0.97	1548	
Weighted Avg	0.97	0.97	0.97	1548	

b) Confusion Matrix of ViT

Besides the classification report, a confusion matrix was prepared to evaluate the class-wise performance of the proposed ViT model on the Onion Dataset, which is illustrated in Figure 4.7 The confusion matrix shows that the diagonal dominance proves the correct classification of most of the samples in all categories. Misclassification rates were very low and were principally due to confusions among visually similar classes, like StripeD and VirosisD, and Healthy_leaf.

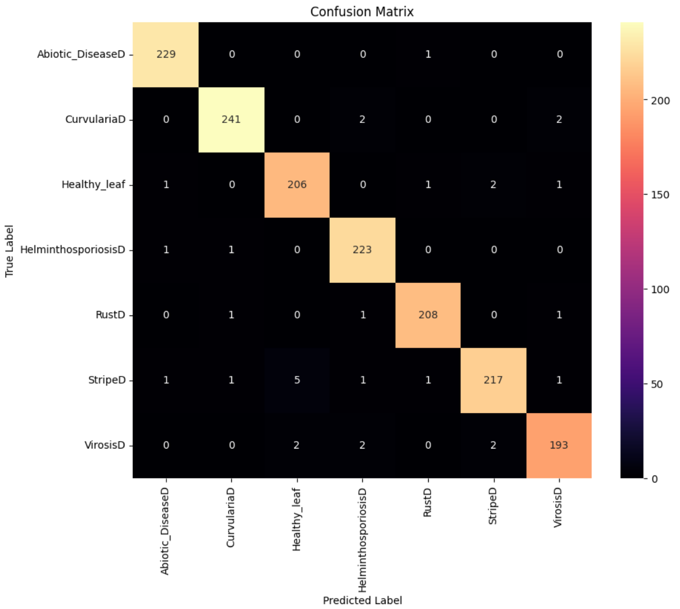


Figure 4.7: Confusion Matrix of Maize

These slight overlaps indicate that the model had some difficulty when disease symptoms were visually similar or looked like healthy leaves. However, the general pattern of confusion matrix shows the high degree of accuracy and robustness of the ViT model to differentiate between onion leaf diseases and supports the findings in Table 4.8

c) Plot Training History of ViT

The history of the training and validation of the proposed Vision Transformer (ViT) model on the Onion Dataset is shown in Figure 4.8. The accuracy curve reveals that it was training more steadily and ultimately settled on 97%, which represents a good learning capacity of the model. Correspondingly, the training and validation loss graphs show a steady decrease and smooth convergence, which represents that the model minimized errors successfully.

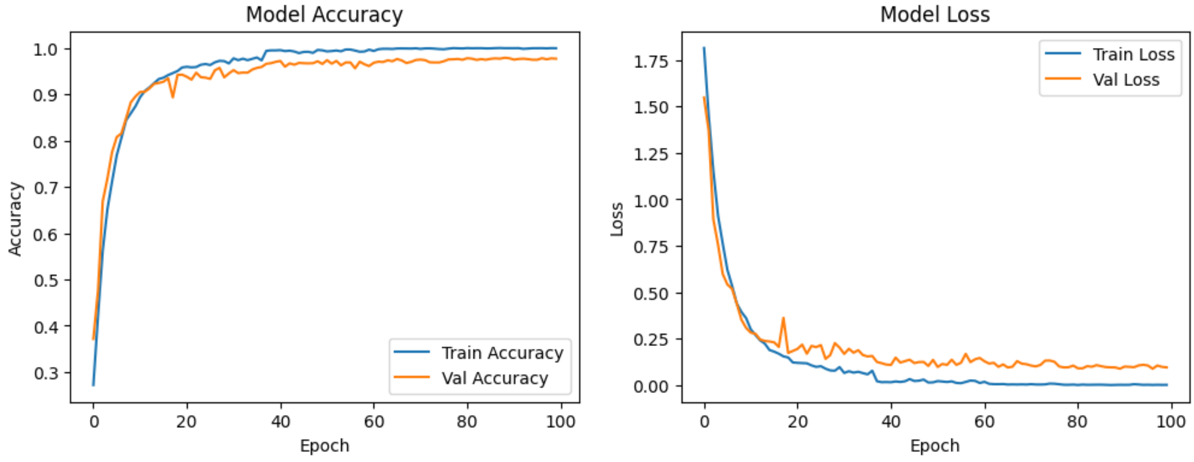


Figure 4.8: Plot Training History of Maize

The similarity of the training and validation curves indicates there was no overfitting or underfitting of the model. This consistent convergence also proves that the ViT model could generalize well to unseen data, and it is in line with the superior classification performances indicated in Table 4.8 and the confusion matrix presented in Figure 4.7.

d) Comparison with Baseline CNN Models

In line with further supporting the effectiveness of the proposed Vision Transformer (ViT) model, performance was compared against baseline CNN-based models. The comparison was made on the basis of classification reports and training history plots of each model.

The ViT model also recorded a high accuracy level of 98%, as shown in Table 4.9, compared to the traditional CNN architectures that recorded an average of 36-96% accuracy. The ViT had a better precision, recall, and F1-scores in the majority of disease types in onion. Specifically, the Abiotic_DiseaseD, CurvulariaD and RustD classes were nearly perfectly classified (approximately 0.99 F1-score), whereas CNN models had a relatively larger proportion of misclassification in visually confusable groups, like StripeD and VirosisD. ViT training and validation history curves, shown in Figure X, are smooth and have low overfitting. Conversely, baseline CNN models were slower to converge and have a larger variance in validation accuracy, signaling poor generalization.

Table 4.9: Classification Report of the Baseline CNN Models

Model	Class	Precision	Recall	F1-Score	Support	Accuracy
ResNet152V2	*Abiotic_DiseaseD*	0.97	0.96	0.96	220	0.96
	CurvulariaD	0.95	0.95	0.95	220	
	Healthy_leaf	0.99	0.99	0.99	220	
	HelminthosporiosisD	0.94	0.95	0.95	220	
	RustD	0.94	0.97	0.95	220	
	StripeD	0.97	0.97	0.97	220	
	VirosisD	0.94	0.92	0.93	220	
DenseNet201	*Abiotic_DiseaseD*	0.99	0.96	0.98	220	0.95
	CurvulariaD	0.92	0.95	0.93	220	
	Healthy_leaf	0.95	0.98	0.96	220	
	HelminthosporiosisD	0.94	0.91	0.93	220	
	RustD	0.94	0.99	0.96	220	
	StripeD	0.95	0.91	0.93	220	
	VirosisD	0.93	0.93	0.93	220	
InceptionResNetV2	*Abiotic_DiseaseD*	0.94	0.95	0.94	220	0.92
	CurvulariaD	0.93	0.94	0.93	220	
	Healthy_leaf	0.95	0.99	0.97	220	
	HelminthosporiosisD	0.87	0.9	0.88	220	
	RustD	0.96	0.95	0.95	220	
	StripeD	0.91	0.87	0.89	220	
	VirosisD	0.87	0.81	0.84	220	
InceptionV3	*Abiotic_DiseaseD*	0.93	0.94	0.93	220	0.92
	CurvulariaD	0.91	0.93	0.92	220	
	Healthy_leaf	0.96	0.97	0.96	220	
	HelminthosporiosisD	0.95	0.89	0.92	220	
	RustD	0.95	0.94	0.94	220	
	StripeD	0.92	0.89	0.9	220	
	VirosisD	0.83	0.89	0.86	220	
ResNet101	*Abiotic_DiseaseD*	0.28	0.72	0.4	220	0.36
	CurvulariaD	0.42	0.68	0.52	220	
	Healthy_leaf	0.47	0.25	0.33	220	
	HelminthosporiosisD	0.47	0.24	0.32	220	
	RustD	0.43	0.19	0.26	220	
	StripeD	0.38	0.27	0.32	220	
	VirosisD	0.23	0.13	0.16	220	
Xception	*Abiotic_DiseaseD*	0.92	0.93	0.93	220	0.91
	CurvulariaD	0.89	0.93	0.91	220	
	Healthy_leaf	0.96	0.98	0.97	220	
	HelminthosporiosisD	0.91	0.9	0.9	220	
	RustD	0.93	0.91	0.92	220	
	StripeD	0.92	0.91	0.92	220	
	VirosisD	0.86	0.84	0.85	220	

All in all, the classification metrics and the training history plots indicate that the proposed ViT model achieved substantially superior results compared to baseline CNN models by offering higher accuracy, greater robustness, and improved generalization over the Onion Dataset.

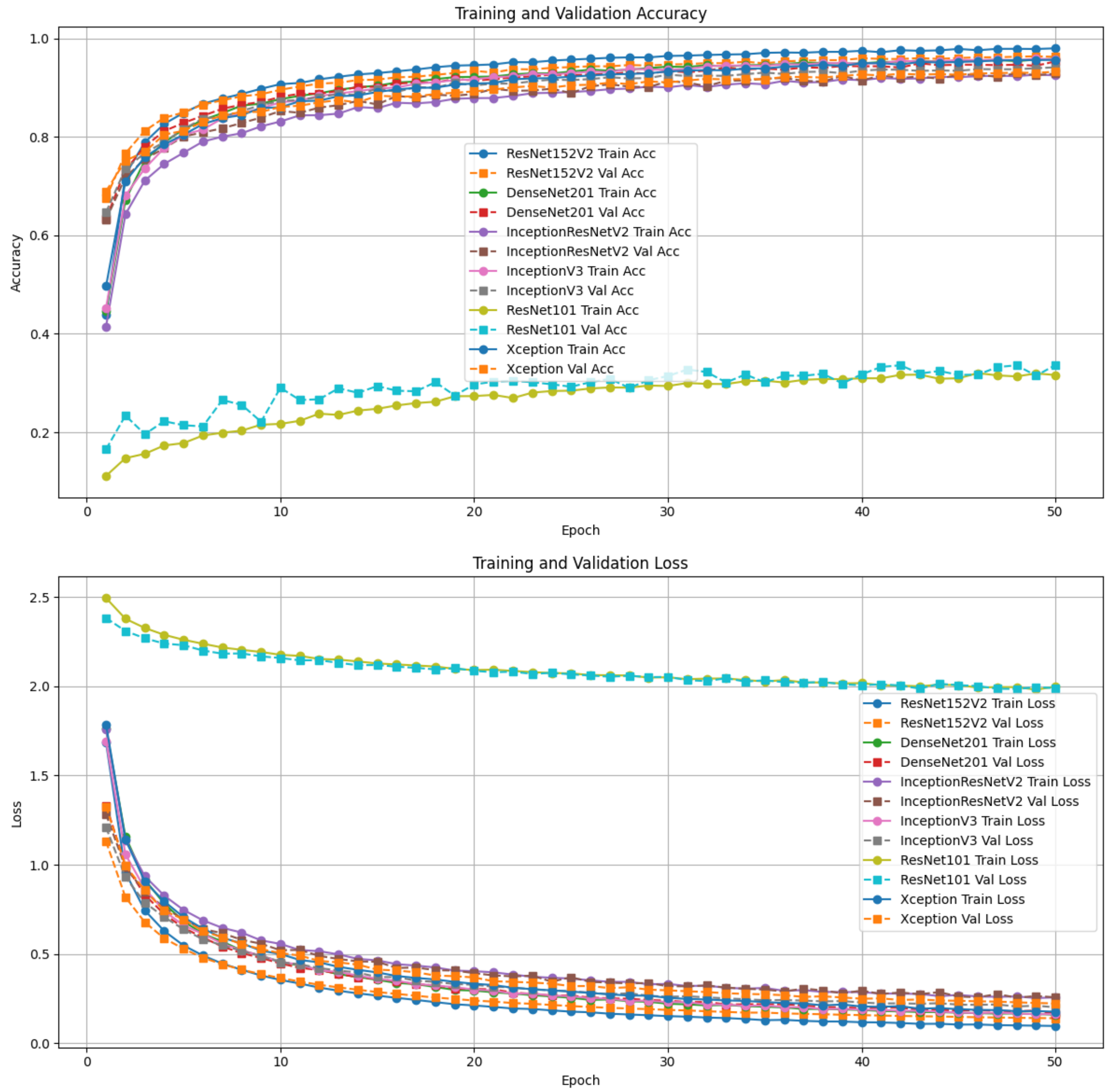


Figure 4.9: Training and validation curves (loss and accuracy) for each fine-tuned transfer learning models for Maize.

4.3.4 Results on Combined TOM Dataset

Having trained the model on separate datasets, it is the time to test the compatibility of the model against all the 23 classes of the Combined TOM dataset. This evaluation establishes whether the model can perform well by processing varieties of categories of diseases at the same time. Also, the learned model weights will be stored and subsequently utilized as alternative weights on the web application to classify real-time images. The performance and efficiency of the model have been presented along with its classification report (CR), confusion matrix (CM), ROC curve, precision-recall curve, and per-class accuracy which gives a detailed analysis of its reliability and effectiveness in the multi-class crop disease detection task.

a) Classification Report

Table 4.10 shows the classification report of the proposed Vision Transformer (ViT) model on Combined TOM dataset. The model displayed an overall accuracy rate of 96% with a high performance on all the 23 classes. The majority of classes, such as Abiotic_DiseaseD, Blossom_end_rot_d, TomL_alternaria_mite_d, and exces_nitrogen_d, showed almost perfect precision and recall, along with F1-scores, indicative of the model to perform well in distinguishing between the various diseases and healthy leaves.

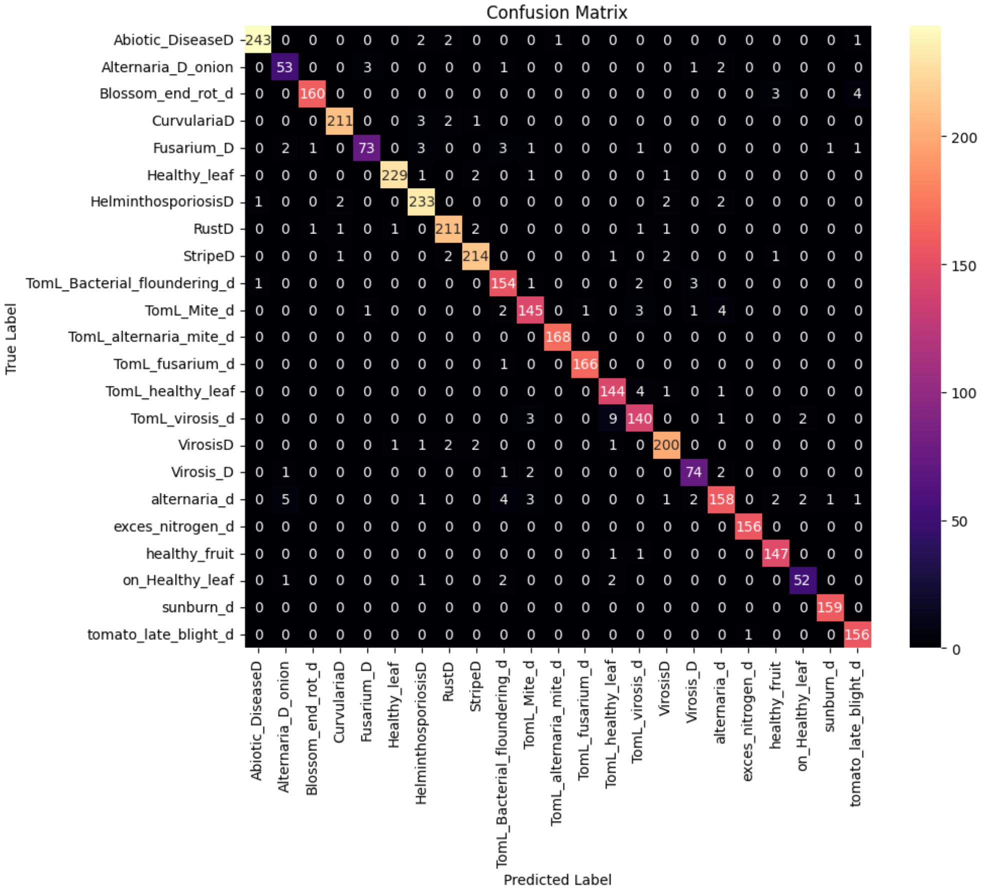
There were some classes with lower scores, e.g., Alternaria_D onion, Fusarium_D, and on Healthy leaf that scored a little lower (F1-scores of around 0.87-0.91), which may be because there is visual similarity of these classes with others. All in all, the macro and weighted averages are high, which affirms that the ViT model can generalize effectively against various classes, giving it reliable results that can be adopted in a web-based crop disease detection system.

Table 4.10: Classification Report of The Combined TOM Dataset

Class	Precision	Recall	F1-Score	Support	Accuracy
Abiotic_DiseaseD	0.99	0.98	0.98	249	0.98
Alternaria_D_onion	0.98	0.98	0.97	60	
Blossom_end_rot_d	0.99	0.99	0.98	167	
CurvulariaD	0.98	0.97	0.98	217	
Fusarium_D	0.98	0.98	0.98	86	
Healthy_leaf	0.99	0.98	0.98	234	
HelminthosporiosisD	0.98	0.98	0.98	240	
RustD	0.97	0.97	0.97	218	
StripeD	0.97	0.97	0.97	221	
TomL_Bacterial_floundering_d	0.99	0.98	0.98	161	
TomL_Mite_d	1.00	0.99	0.99	157	
TomL_alternaria_mite_d	1.00	1.00	1.00	168	
TomL_fusarium_d	0.99	0.99	0.99	167	
TomL_healthy_leaf	0.98	0.98	0.98	150	
TomL_virosis_d	1.00	0.99	0.99	155	
VirosisD	0.98	0.97	0.97	207	
Virosis_D	0.99	0.99	0.99	80	
alternaria_d	0.99	0.98	0.98	180	
exces_nitrogen_d	1.00	1.00	1.00	156	
healthy_fruit	0.99	0.99	0.97	149	
on_Healthy_leaf	0.93	0.90	0.91	58	
sunburn_d	1.00	1.00	0.99	159	
tomato_late_blight_d	0.99	0.99	0.99	157	
Macro Avg	0.98	0.98	0.98	3796	
Weighted Avg	0.98	0.98	0.98	3796	

b) Confusion Matrix

The confusion matrix illustrating the class-wise performance of the model across the various classes of the Combined TOM dataset is displayed in Figure 4.10. Most of the predictions are on the diagonal and this implies that most samples are correctly classified. Small misclassifications were detected in similar-looking classes, e.g. AlternariaD_onion vs. FusariumD and TomL_virosisd vs. TomLhealthy_leaf. In general, the confusion matrix proves the strength of ViT model in separating various disease categories as well as justifying the high accuracy displayed in the classification report.



True Label \ Predicted Label	Abiotic_DiseaseD	Alternaria_D_onion	Blossom_end_rot_d	CurvulariaD	Fusarium_D	Healthy_leaf	HelminthosporiosisD	RustD	StripeD	TomL_Bacterial_floundering_d	TomL_Mite_d	TomL_alternaria_mite_d	TomL_fusarium_d	TomL_healthy_leaf	TomL_virosis_d	VirosisD	Virosis_D	alternaria_d	exces_nitrogen_d	healthy_fruit	on_Healthy_leaf	sunburn_d	tomato_late_blight_d
Abiotic_DiseaseD	243	0	0	0	0	0	2	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
Alternaria_D_onion	0	53	0	0	3	0	0	0	0	1	0	0	0	0	0	0	1	2	0	0	0	0	0
Blossom_end_rot_d	0	0	160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	4
CurvulariaD	0	0	0	211	0	0	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fusarium_D	0	2	1	0	73	0	3	0	0	3	1	0	0	0	1	0	0	0	0	0	0	1	1
Healthy_leaf	0	0	0	0	0	229	1	0	2	0	1	0	0	0	0	1	0	0	0	0	0	0	0
HelminthosporiosisD	1	0	0	2	0	0	233	0	0	0	0	0	0	0	0	2	0	2	0	0	0	0	0
RustD	0	0	1	1	0	1	0	211	2	0	0	0	0	0	1	1	0	0	0	0	0	0	0
StripeD	0	0	0	1	0	0	0	2	214	0	0	0	0	1	0	2	0	0	0	1	0	0	0
TomL_Bacterial_floundering_d	1	0	0	0	0	0	0	0	0	154	1	0	0	0	2	0	3	0	0	0	0	0	0
TomL_Mite_d	0	0	0	0	1	0	0	0	0	2	145	0	1	0	3	0	1	4	0	0	0	0	0
TomL_alternaria_mite_d	0	0	0	0	0	0	0	0	0	0	0	168	0	0	0	0	0	0	0	0	0	0	0
TomL_fusarium_d	0	0	0	0	0	0	0	0	0	1	0	0	166	0	0	0	0	0	0	0	0	0	0
TomL_healthy_leaf	0	0	0	0	0	0	0	0	0	0	0	0	0	144	4	1	0	1	0	0	0	0	0
TomL_virosis_d	0	0	0	0	0	0	0	0	0	0	3	0	0	9	140	0	0	1	0	0	2	0	0
VirosisD	0	0	0	0	0	1	1	2	2	0	0	0	0	1	0	200	0	0	0	0	0	0	0
Virosis_D	0	1	0	0	0	0	0	0	0	1	2	0	0	0	0	0	74	2	0	0	0	0	0
alternaria_d	0	5	0	0	0	0	1	0	0	4	3	0	0	0	0	1	2	158	0	2	2	1	1
exces_nitrogen_d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	156	0	0	0	0
healthy_fruit	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	147	0	0	0
on_Healthy_leaf	0	1	0	0	0	0	1	0	0	2	0	0	0	2	0	0	0	0	0	0	52	0	0
sunburn_d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	159	0
tomato_late_blight_d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	156

Figure 4.10: Confusion Matrix of Combined TOM Dataset

c) Plot Training History

The training and validation history, shown in Figure 4.11 indicates that the model converged well without any jitter. Training and validation accuracy curves increase steadily and shoot to about 98%, whereas the loss curves decrease steadily. The similarity between the training and validation curves indicates that there is insignificant overfitting, which is evidenced by the fact that the ViT model has a good generalization ability to unseen data. This consistent convergence indicates effectiveness and consistency of the model in training.

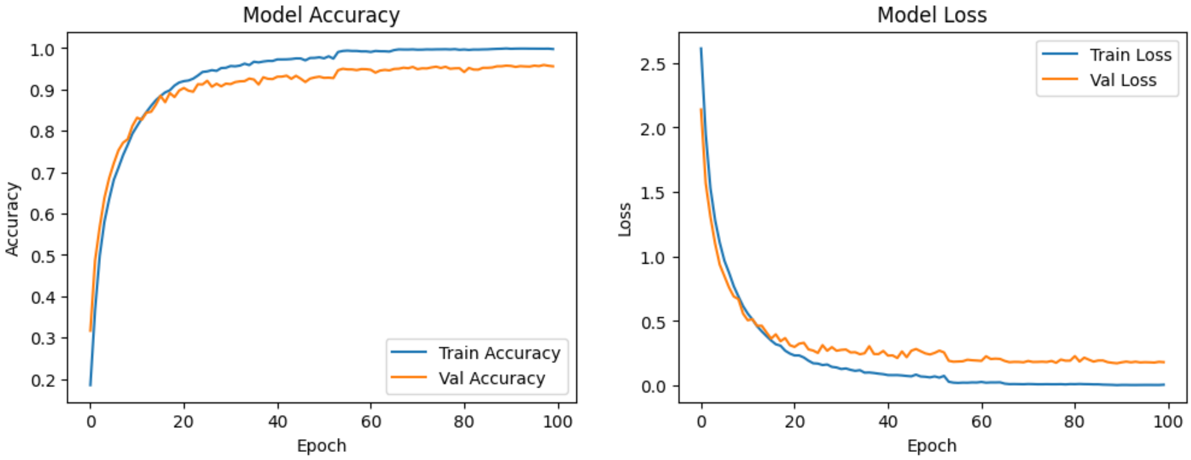


Figure 4.11 Plot Training History of TOM

d) Roc-Curve

Figure 4.12 represents a ROC curve which tests the false positive against the true positive rate per class. The majority of classes demonstrate the area under the curve (AUC) near to 1.0, which suggests a perfect discriminative power of the model regarding all of the disease groups. This validates that not only does the ViT model offer good accuracy but also good class separability and thus suited to multi-class crop disease detection.

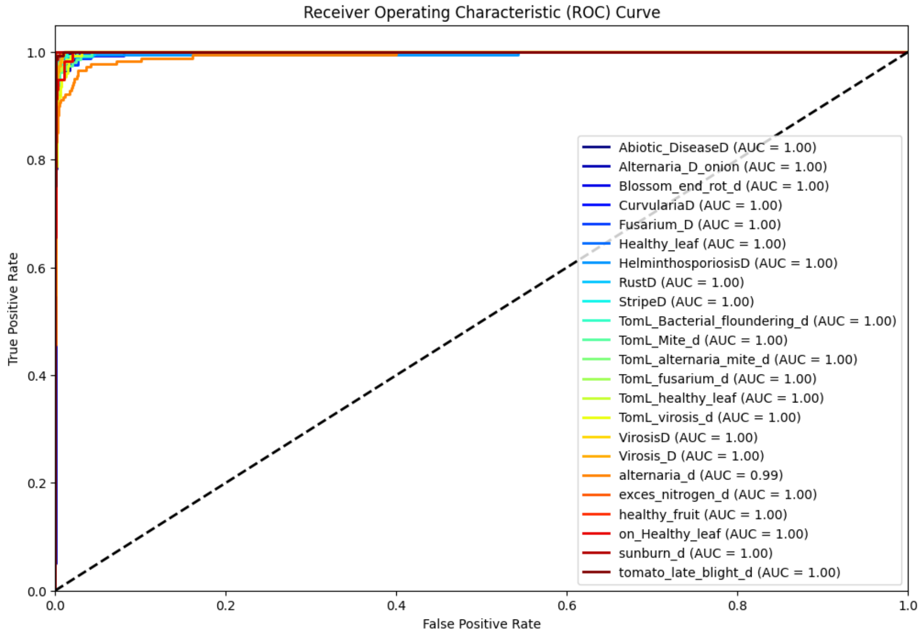


Figure 4.12: ROC-Curve TOM

4.3.5 Ablation Study

Following the development of the proposed model an ablation study has been conducted in order to examine the potential effect of the various configurations on the performance of the model in question. The overall idea was to determine, in this case, whether any changes in the design or setup might lead to results that are equal to or better than the default configuration of the proposed model. Table 4.11 was used to study the impact of changing the size of the input image and patch to see how changing the size of an image influences the extraction of features and a classification accuracy in terms of spatial resolution.

Table 4.11: Ablation study of the proposed model with different patch and image sizes.

Patch Size	Image Size	Parameters	Accuracy
8	128	2,402,186	97.37%
8	256	2,402,186	97.44%
16	128	2,475,914	97.31%
16	256	2,475,914	97.11%
32	**128**	**2,770,826**	**98.70%**
32	256	2,770,826	97.24%

Then, in Table 4.12, the number of transformer layers, attention heads, and hidden dimensions were changed to observe how architectural depth and attention complexity influenced performance.

Table 4.12: Ablation study of the proposed model by modifying transformer layers, number of heads, and hidden dimensions.

Transformer Layers	Number of Heads	Hidden Dimension	Parameters	Accuracy
6	**4**	**128**	**2,770,826**	**98.70%**
7	8	128	5,012,618	97.11%
6	6	64	994,506	97.04%
8	8	128	5,672,330	97.44%
8	6	192	10,072,138	96.91%
4	4	128	1,978,762	97.11%

Furthermore, in Table 4.13, different activation functions (such as ReLU, Sigmoid, Softmax, and Linear) were tested to evaluate their impact on both accuracy and loss values.

Table 4.13: Ablation study of the proposed model using different activation functions and their impact on accuracy and loss.

Activation	Train Acc	Val Acc	Test Acc	Train Loss	Val Loss	Test Loss
Softmax	**99.97%**	**98.84%**	**98.70%**	**0.0130**	**0.0650**	**0.0790**
Sigmoid	99.20%	96.81%	97.11%	0.0148	0.0977	0.0935
ReLU	99.17%	96.94%	97.44%	0.0163	0.0952	0.0944
Linear	98.75%	96.74%	97.04%	0.0267	0.1177	0.0948

Finally, Table 4.14 presents a comparison of multiple optimizers (like Adam, SGD, RMSprop, Adagrad, AdamW, and Adamax) to determine which one best supported the model's learning process.

Table 4.14: Ablation study of the proposed model using different optimizers and their impact on accuracy and loss.

Optimizer	Train Acc	Val Acc	Test Acc	Train Loss	Val Loss	Test Loss
Adam	**99.97%**	**98.84%**	**98.70%**	**0.0130**	**0.0650**	**0.0790**
SGD	99.15%	96.34%	96.52%	0.0154	0.1413	0.1342
RMSprop	99.18%	97.07%	97.44%	0.0148	0.1023	0.0830
Adagrad	59.79%	62.27%	64.06%	1.1010	1.0334	1.0208
AdamW	99.23%	96.94%	97.17%	0.0141	0.0904	0.0880
Adamax	98.68%	96.91%	97.44%	0.0307	0.1011	0.0829

This systematic exploration was done to validate whether the originally proposed settings were indeed optimal or if any alternative configuration could yield improved results. From all these insights, it is indeed can be stated that, originally proposed settings were a suitable one as it excels all the other settings in notable performance

4.3.6 Interpretation with XAI

The proposed Vision Transformer (ViT) model needs trust and clarity that is why three explainable AI (XAI) techniques (SHAP and LIME) have been applied to interpret the model decisions and visualize them. This section also includes visual explanations in form of attention-based heat-maps and pseudo-segmentation masks.

a) SHAP

SHAP (Shapley Additive Explanations) is employed to gain knowledge about the contribution of each pixel area to the model prediction (Mary & Kavita 2024). Unlike attention maps, which are visibly indicative of greater regions of focus or pseudo-segmentation masks, which is representative of a potential area of disease, values in SHAP also specially subtract or add the pixel contribution to the resulting confidence of the model. The visualization of SHAP values in figure 4.15 will be an overlay mask on the original picture and in such a manner, the red colored area will signify positive to the predicted type whereas the blue colored area will signify a negative impact. The approach offers pixel-resolution explanations of high granularity that enables describing the image components that favored and deteriorated the trust of the model in the outcome of the prediction. SHAP strengthens the transparency of models and increases explainability by defining the importance of pixels.

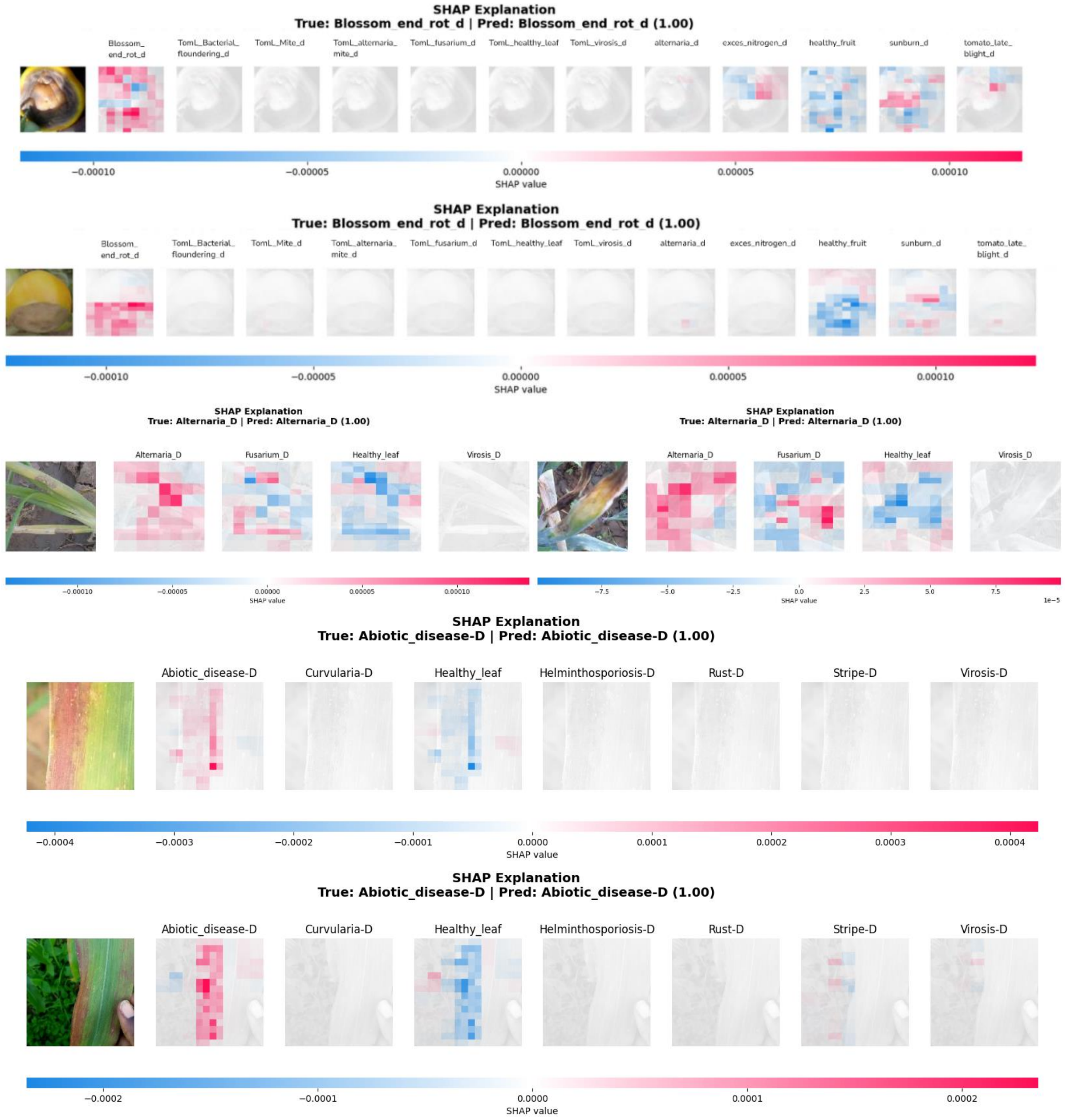


Figure 4.13: SHAP Analysis for Tomato, Onion, Maize

b) LIME

Local Interpretable Model-agnostic Explanations (LIME) is a methodology of interpreting model predictions at a local level by approximating them using an interpretable surrogate model (Ribeiro et al., 2016). In images, LIME breaks the input into superpixels and perturbs them to see their impact on the prediction outcome. The visualization shows the superpixels that had the biggest impact on the classification and therefore it is easy to see what parts of the leaf image the model used. Such localized explanation enhances interpretability, transparency, and credibility in the predictions of the system.

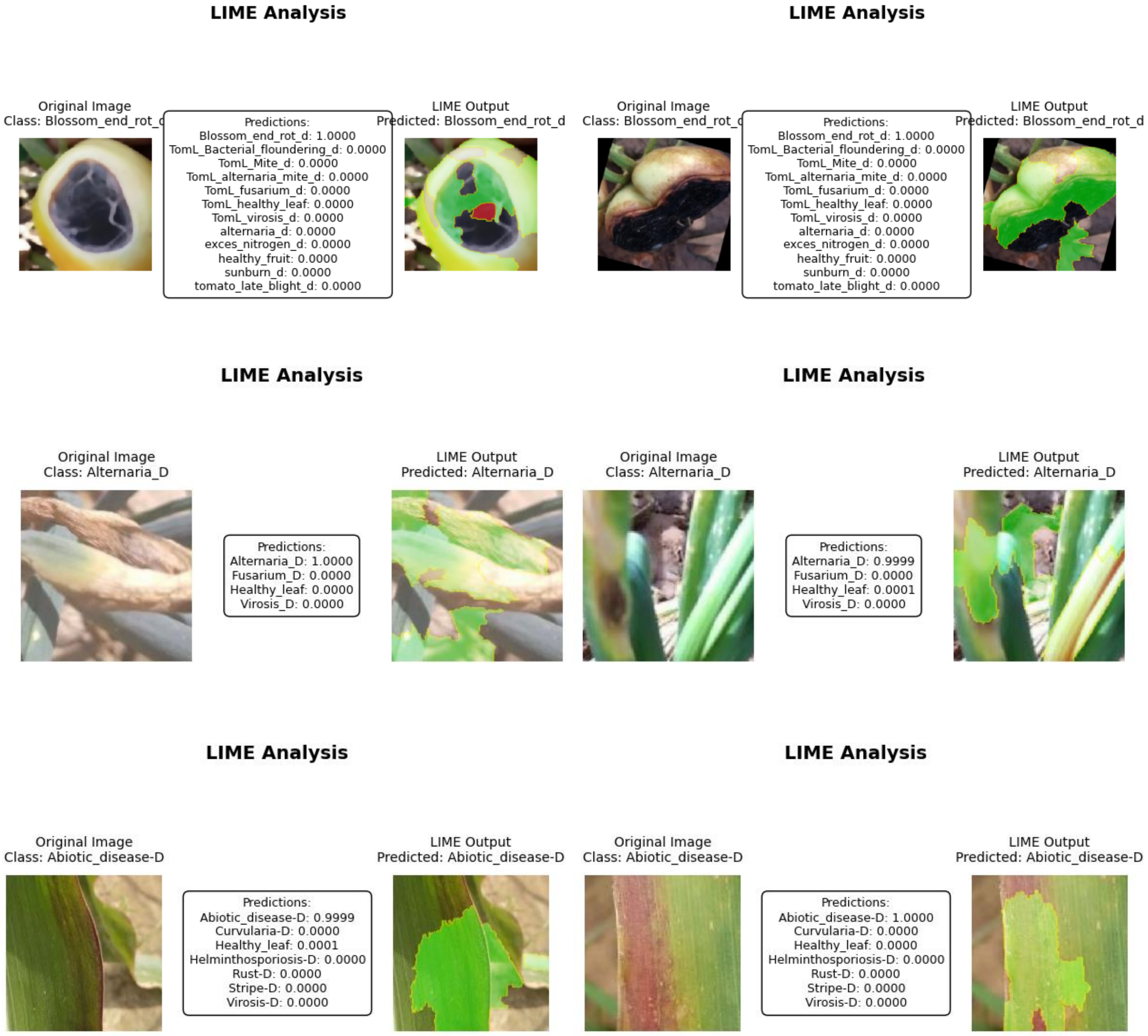


Figure 4.14: LIME Visualization for Tomato, Onion, and Maize

c) Pseudo Segmentation

To maximize interpretability and facilitate the transition between the outputs of such classification tasks and clinical-level relevance, a novel pseudo-segmentation XAI method that is intended to highlight areas in fundus photographs that are considered pathology-specific and pathological is proposed. Even though transformer attention maps provide top-level visual attention, they do not provide pixel-level specific localization and quantification. The proposed approach overcomes this by producing pseudo-segmentation masks and it utilizes HSV color thresholding, concatenated with morphological dilation. It attacks the common retinal abnormalities which include red spots (hemorrhages), and yellow-white exudates. These features are detected by the algorithm (using specific HSV ranges), and the areas are segmented and binary masks are generated that demarcate the affected areas. The technique also computes quantitative measures of lesion area percentage, mean RGB value of diseased areas that provide quantitative measures of pathology severity on a per test image basis. Figure 4.16 demonstrates each sample, the original image (with true/predicted classes and probability of the top-3 classes), the generated attention heatmap with a soft attention focusing, and the pseudo-segmented mask. The black mask of the background is designed to show non-informative regions, and color-coded regions (orange, red, brown) show potential regions of lesions. This type of layered visualization framework is capable not only of confirming that the model is attending clinically relevant area but also of emulating diagnosis support environment. It enables clinicians to know where, how much and what is impacted, hence enhancing model trust, transparency and utility within real-life screening workflow.

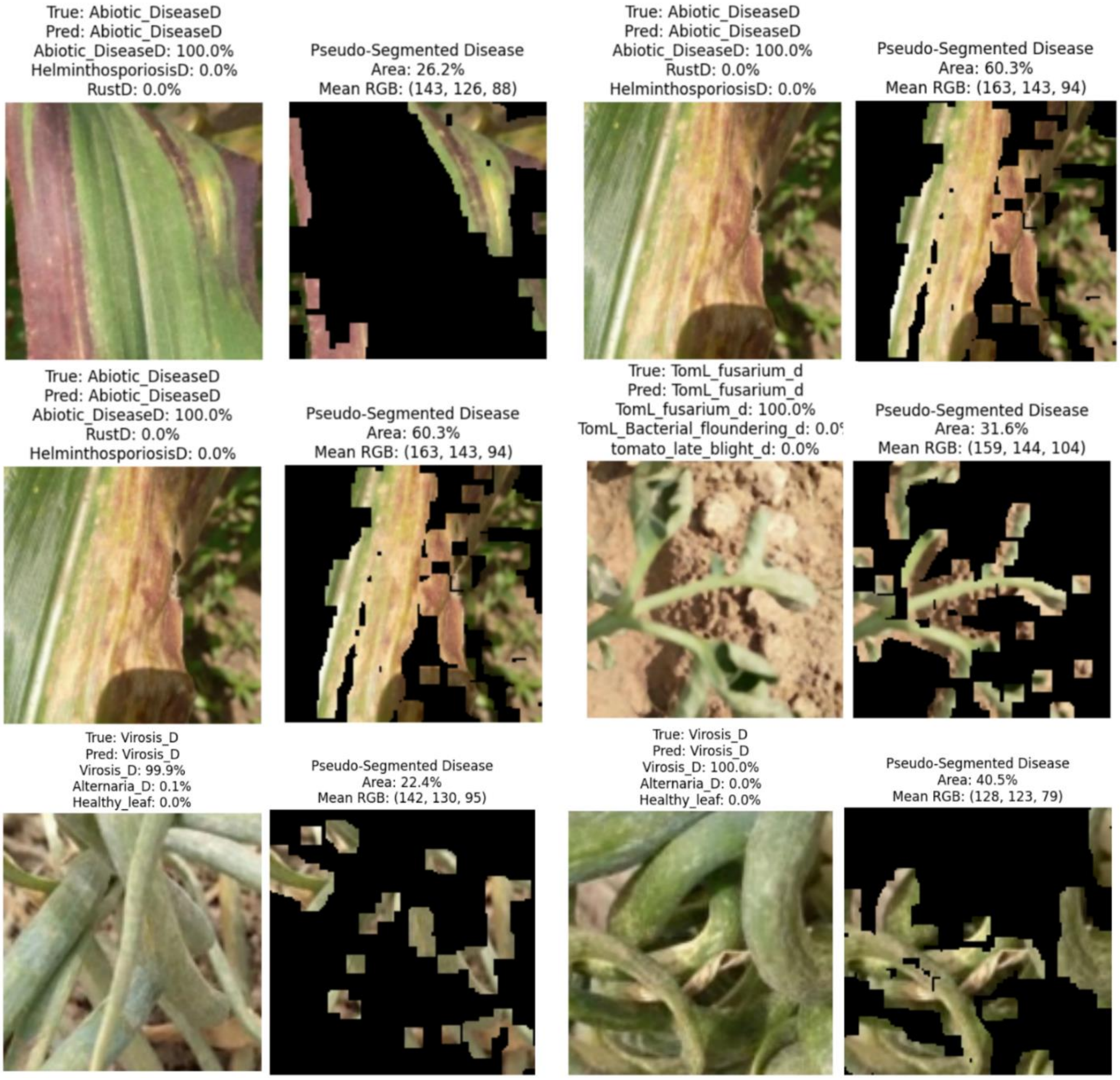


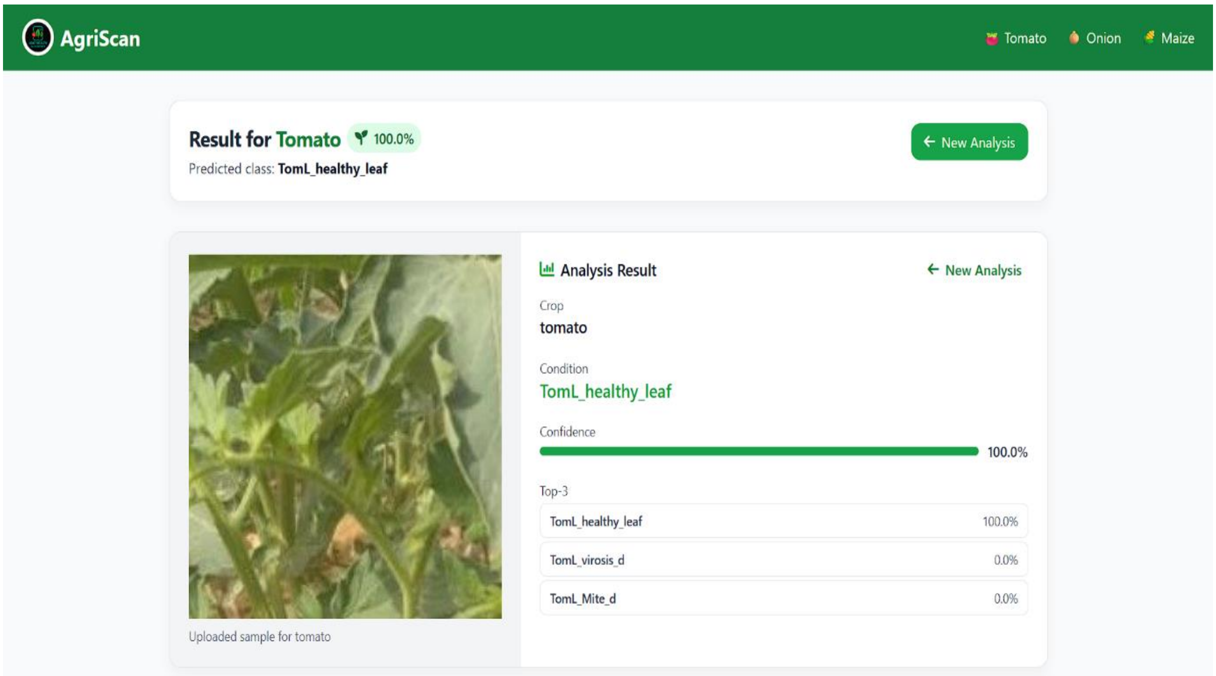
Figure 4.15: Pseudo Segmentation Visualization for Tomato, Onion, and Maize

4.3.7 Web Application Deployment

This work developed a web-based application called AgriScan and deployed the same to enable automated prediction and reporting of the health condition of plants on a leaf level. It was proposed that the end-users (e.g., farmers or agricultural specialists) would be able to upload an image of a leaf on their devices and that the image would be processed by the deployed model. After the processing is finished, the application presents a definite classification of the state of the leaf and additional information and reports. The deployment results are listed below.

Prediction Result and Classification Display

During the first phase, the user will be able to choose and upload any picture of a plant leaf on his or her device. The system recognizes the related class (e.g., TomL_healthy_leaf) after analysis and presents it on the interface directly. In case the image is associated with a known and determinable condition, the corresponding predicted class is indicated in green text. On the contrary, when the uploaded image is not reliably classified, the system will present the prediction in red text or markers to show that they could not identify it. This makes it clear to the user on whether the prediction holds water.



Result for Tomato 100.0%
Predicted class: TomL_healthy_leaf

Crop
tomato

Condition
TomL_healthy_leaf

Confidence
100.0%

Top-3

TomL_healthy_leaf	100.0%
TomL_virosis_d	0.0%
TomL_Mite_d	0.0%

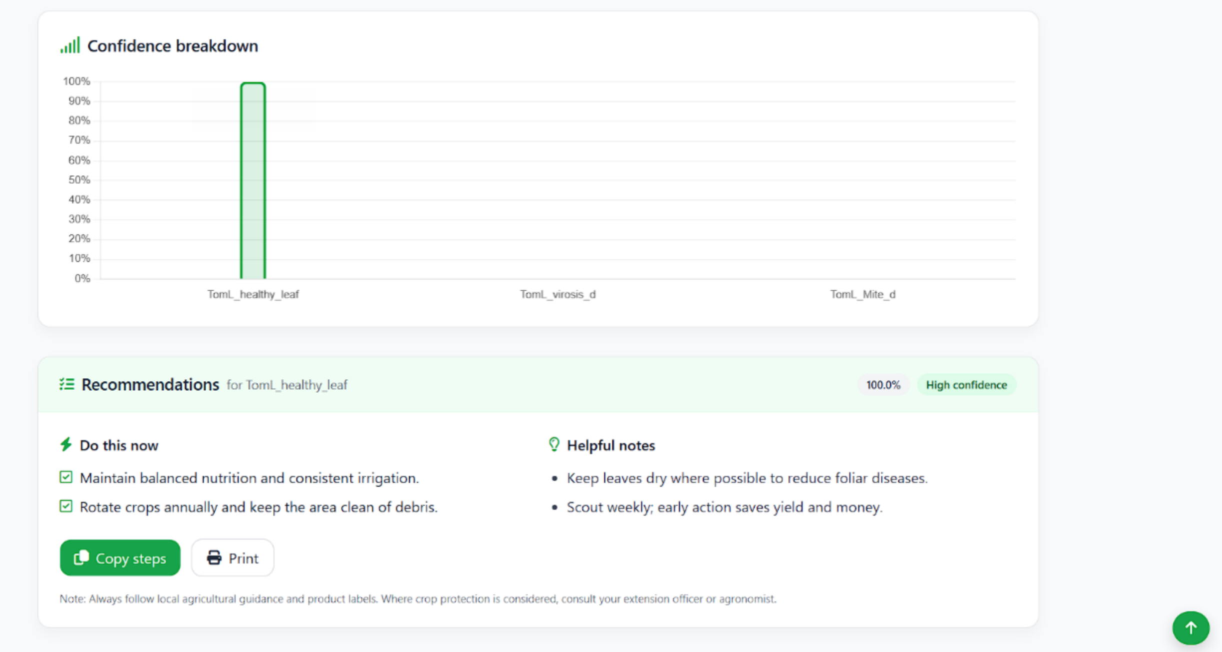
Uploaded sample for tomato

Figure 4.16: Displaying Prediction Result

Confidence Breakdown and Visualization

After the classification, the system offers a bar graph, which shows a confidence chart. This shows the confidence level of the model to the three most probable classes.

As an example, when the model is 100 percent certain that the leaf is healthy, only one bar indicates the complete certainty. In case of more competing predictions, the system displays the distribution of the confidence percentages over up to three classes. This visualization makes possible the interpretation of not only the final decision but also the relative certainty of the model.



Recommendations for TomL_healthy_leaf 100.0% High confidence

Do this now

☑ Maintain balanced nutrition and consistent irrigation.
☑ Rotate crops annually and keep the area clean of debris.

Helpful notes

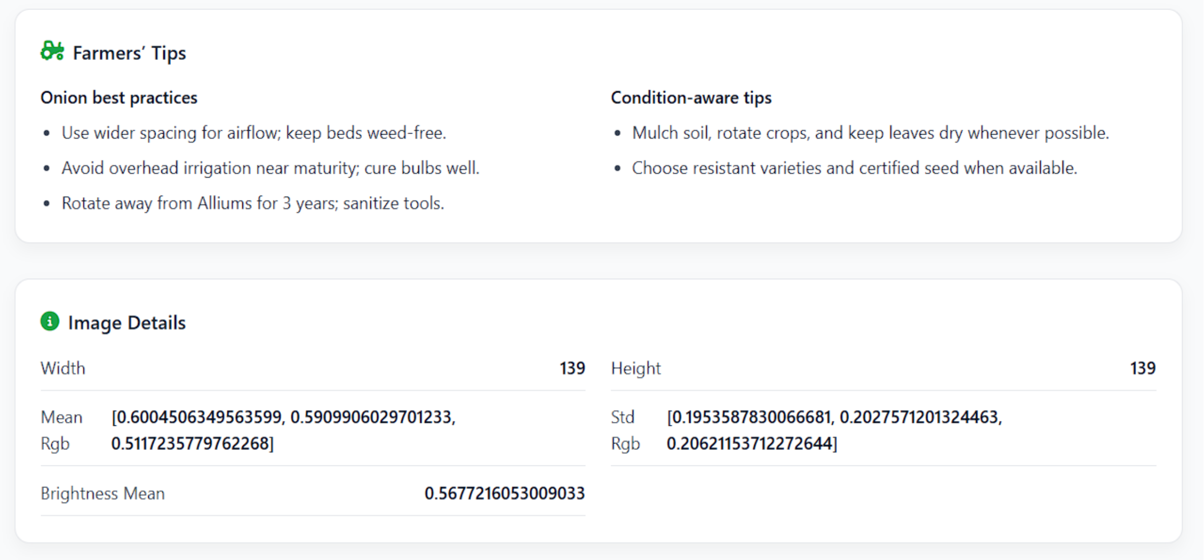
- Keep leaves dry where possible to reduce foliar diseases.
- Scout weekly; early action saves yield and money.

Note: Always follow local agricultural guidance and product labels. Where crop protection is considered, consult your extension officer or agronomist.

Figure 4.17: Confidence Chart with Suggested Actions (Recommendation)

Image Details and Condition-Specific Guidance

Other metadata of the images, including width, height, RGB mean values, standard deviations, and brightness mean, are also presented in the web application. The information offers a piece of technical evidence of validation and future studies. Together with it, the system produces condition-conscious tips and best practice recommendations. An example is with healthy tomato leaves, where we recommend staking or trellising to encourage air circulation, dropping lower leaves that contact the ground, and keeping fertilization and watering balanced. These insights fill the gap between theory and practice of technical prediction and practical field advice to farmers.



Farmers' Tips

Onion best practices

- Use wider spacing for airflow; keep beds weed-free.
- Avoid overhead irrigation near maturity; cure bulbs well.
- Rotate away from Alliums for 3 years; sanitize tools.

Condition-aware tips

- Mulch soil, rotate crops, and keep leaves dry whenever possible.
- Choose resistant varieties and certified seed when available.

Image Details

Width	139	Height	139
Mean Rgb	[0.6004506349563599, 0.5909906029701233, 0.5117235779762268]	Std Rgb	[0.1953587830066681, 0.2027571201324463, 0.20621153712272644]
Brightness Mean	0.5677216053009033		

Figure 4.18: Farmer Tips and Image Details

Explainability of Predictions

To increase the reliability and explainability of the used system, the three explainability methods were added to the application. The former is LIME Superpixel Explanation that emphasizes the local regions of the image, called superpixels, whose contribution to the classification decision was the greatest. Highlighting these areas of influence, LIME enables the user to visually verify whether the model is attending to meaningful biological characteristics of the leaf, e.g., veins, color patches, or texture.

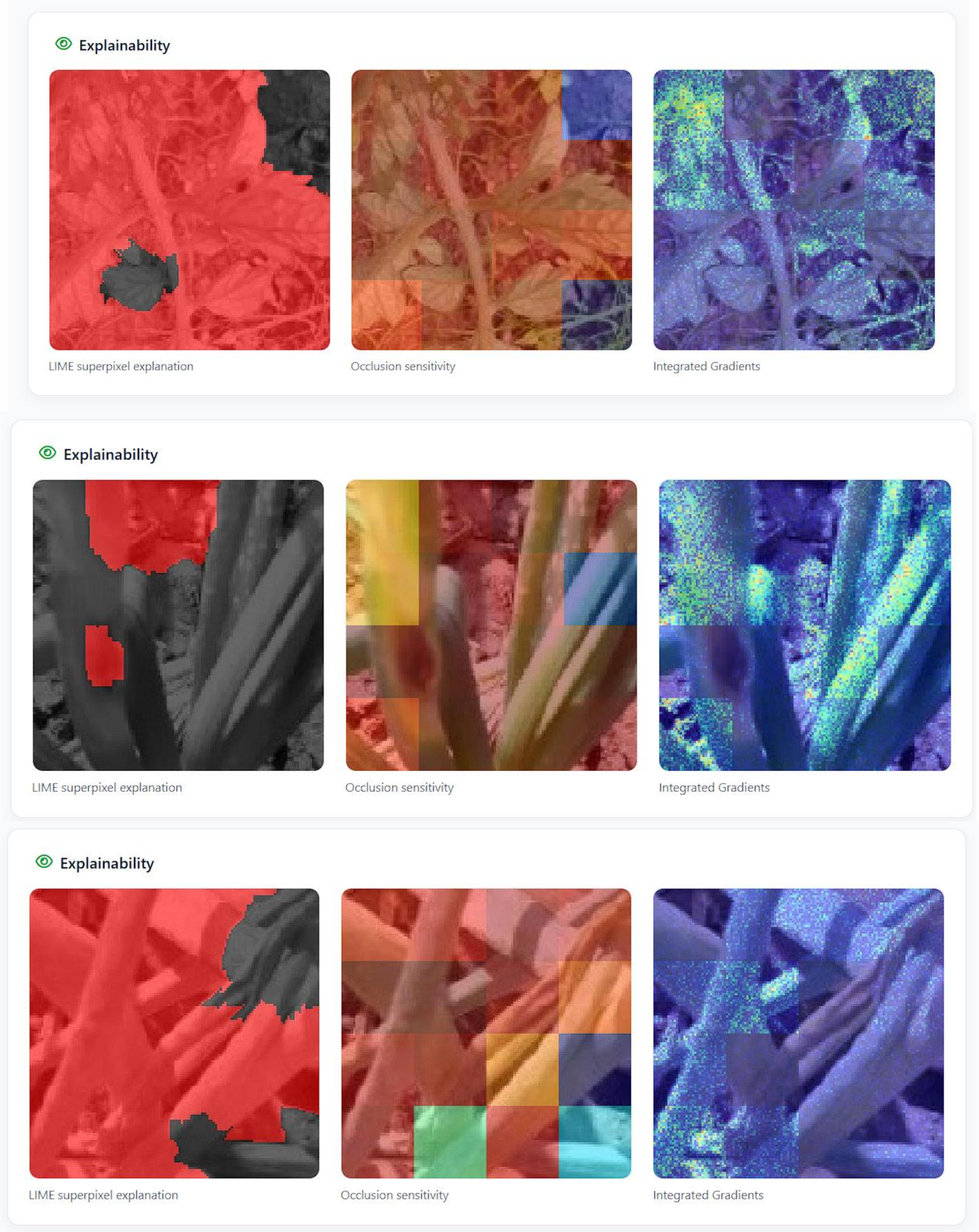
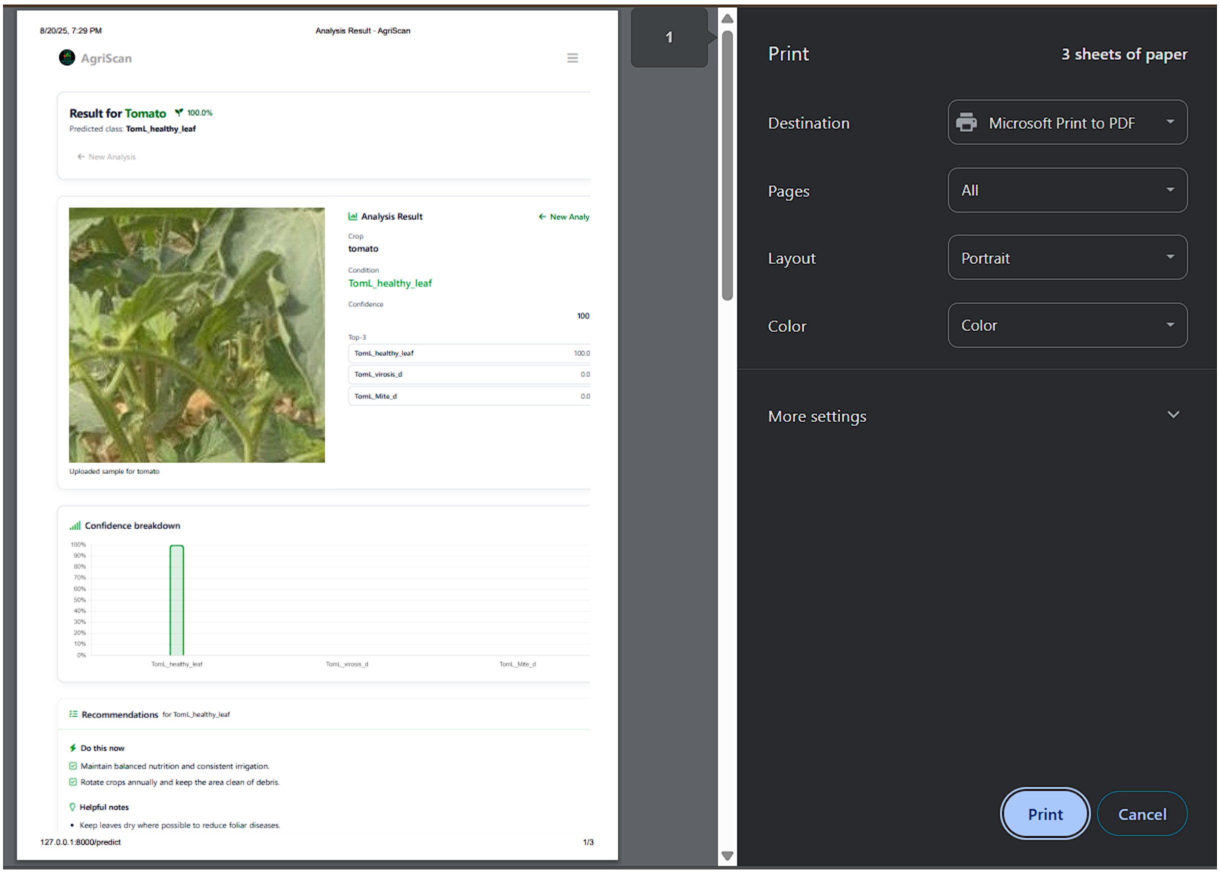


Figure 4.19: Explainability of the Predicted Different Leaf Images

The second technique is Occlusion Sensitivity that operates under the principle of masking various regions of the picture and tracking the modifications in the prediction confidence. This process can assist in determining the areas that are critical in the decision-making of the model because a major decline in the confidence level shows that the occluded region is important. The third approach is Integrated Gradients that generates pixel-level attributions by computing how each pixel contributes to the ultimate prediction. The method is used to visualize feature importance in the whole picture, providing more information about the influence of fine-grained details on the outcome. All these three explainability modules allow making the reasoning process of the deep learning model more transparent, and users can understand why a specific prediction was made and that the system cannot be discussed as a black box.

Report Generation and Printing

Lastly, the app facilitates creation of the overall report that merges the uploaded image, the results of prediction, confidence breakdown, recommendations, and explainability visualization. The user can export to PDF and print this report. Farmers and experts can properly document their findings, off-line sharing of the findings and employ the reports in making agricultural decisions or consulting the reports further with such a feature.


8/20/25, 7:29 PM
Analysis Result - AgriScan

Result for Tomato 100.0%
Predicted class: **TomL_healthy_leaf**

Analysis Result
Crop
tomato
Condition
TomL_healthy_leaf
Confidence
100

Top-3
TomL_healthy_leaf 100.0
TomL_virosis_d 0.0
TomL_Mite_d 0.0

Uploaded sample for tomato

Recommendations for TomL_healthy_leaf

Do this now
☑ Maintain balanced nutrition and consistent irrigation.
☑ Rotate crops annually and keep the area clean of debris.

Helpful notes
- Keep leaves dry where possible to reduce foliar diseases.

127.0.0.1:8000/predict
1/3

Print
3 sheets of paper
Destination: Microsoft Print to PDF
Pages: All
Layout: Portrait
Color: Color

Figure 4.20: Interface of Printing the Generated Image Report.

4.4 Summary

The chapter demonstrated the application and outcome of the proposed deep learning model of multi-crop leaf disease classification, which included tomato, onion, maize, and a combined dataset. The paper outlined the computational environment, including both cloud-based and local infrastructures, as well as pre-processing, including image normalization, resizing, augmentation, and class balancing. Optimization techniques that were used during training were early stopping, checkpointing, and learning rate scheduling to achieve convergence and avoid overfitting. The performance was evaluated with the help of various metrics- precision, recall, F1-score, and accuracy, to give a sound assessment. ViT was able to perform better than the six baseline CNN models across datasets with an accuracy range of 96-98 percent. Although ViT performed almost perfectly in all categories, slight misclassifications were noticed in similar classes that are hard to tell apart visually, e.g., healthy leaves and mild infections. In order to increase transparency, SHAP, LIME and pseudo-segmentation (Explainable AI, or XAI) algorithms, which offered both local and global interpretability, were implemented. These methods emphasized disease-affected areas in leaf images, which helped to reinforce the predictability of such results and discussed the issue of the disjunction between AI models and real farming. In conclusion, the framework has been implemented successfully as AgriScan web application. The system facilitates disease prediction, visualization of the confidence, explainability features, and farmer-specific recommendations, and allows generating reports that can be used in practice.

Chapter 5

Engineering Standards and Design Challenges

The standards of engineering met during the development process of the crop leaf disease detection system especially tomato, onion and maize crop is outlined in this chapter. It also reviews the wider, societal, environmental, and other sustainability effects of the project. It was applied through a strict adherence to the existing rules of software engineering, hardware compatibility, and the communication steps as applicable to ascertain the reliability and compliance of the system. Besides, ethical requirements, such as data privacy, model transparency, and equity of access-- were considered to comply with responsible AI practices. The project is driven not just by the need to present a technical solution that provides high performance, but also add a positive impact to sustainability with regards to agriculture, food security and livelihood of farmers in developing countries.

5.1 Compliance with the Standards

These efforts have been based on established engineering practices of software development, data handling and model deployment. This project uses generally agreed upon engineering standards to achieve compatibility, reproducibility, and reliability. The overall system was conceived with emphasis on ethical and technical conformity, high degree of system functioning, and its compatibility with sustainable processes. It was developed on the basis of the open-source platforms and tools that are recognized as widely accepted in machine learning research area. Google colab and kaggle were selected as the main development and experimentation platforms and both of them support GPU acceleration and allow using the standardized execution environments to conduct deep learning work. It allows repeatability and scalability of experiments between systems. The experimentation and model test rig local system was a Victus by HP Laptop; high performance, needed to run the active tests; powered by an AMD Ryzen 7 5800H processor, DDR4 16GB RAM, 2 GPUs, NVidia GeForce RTX 3060 (Laptop) and AMD Radeon Graphics. Windows 11 Home is an operating system compatible with Microsoft (Version 10.0.26100) and provided stability and security to the train a model and pre-processed data.

Standard image resizing and batch processing tools like Microsoft PowerToys were employed to make sure that all input images of different sets had consistent dimensions. All software dependencies such as TensorFlow, PyTorch and relevant libraries were chosen according to their active maintenance and adherence regarding general ML development practices. The usage of open-source software, the references to the compatibility of the platforms and the transparency of the reporting are also indicators of the adherence of this project to the existing standards of engineering and research. These choices improve the reproducibility, sustainability and prospects of field implementation, particularly in resource-limited agricultural settings, of the project.

5.1.1 Software Standards

In developing the software aspect of this project, it was done in accordance with the internationally accepted standards in engineering so as to give quality, reliability and maintainability of the entire system. In particular, the ISO/IEC 25010:2011 has been pursued as the guiding standard regarding the software product quality. Among features that are to be considered in this standard are functionality, performance-efficiency, compatibility, usability, reliability, security, maintainability and portability. Following these requirements made the system be robust, modular, and scalable towards future extension and development in implementation in real situations of farming.

The flexibility and sustainability of the structure was conceived in the whole building structure. Through the usage of free, popular and well-supported tools and environments, the project has maximum accessibility and sustainability. Besides, all elements, including model training or user interface, were chosen depending on their capacity to follow best practices in contemporary software engineering.

- **Model Development and Experimentation:** The model was developed in Python, which is one of the most popular languages pertinent to data science and artificial intelligence disciplines. All the code has been produced in conformity with PEP 8 style standards to facilitate code clarity, easiness in maintenance and collaboration. TensorFlow and Keras machine learning frameworks were used to create and train convolutional neural network and transformer-based machine learning models to perform crop disease classification. The NumPy and Pandas libraries made numerical computing and data manipulation easy and OpenCV was used to process images and augment them. The main place of development and experimentation was on the cloud-based platform such as Google Colab and Kaggle. They had access to high-performance graphics cards and preconfigured ML environments, speeding up the model training cycle dramatically as well as creating no complex steps to install locally. Being collocative and scalable, they also made reproducibility and simple experimentation with various configuration of datasets and variants of models easy

- **Data preparation and Preprocessing Tools:** At the data preparation stage, the Microsoft PowerToys Image Resizer utility was applied in order to ensure consistency in image resolution and file structure in a massive and unbalanced data. It was done to ensure that all images match with the desired models as regards to the intended input dimensions. The directories, filenames and files were uniform across basic Windows file system standards reducing the potential of error at runtime once the data is being consumed. Along with that, augmentation mechanisms such as rotation, flipping, zooming, and brightness modification were used to synthetically balance the distributions of the X-ray classes and enhance the generalization performance of the model by means of Keras and OpenCV.

- **Logitech For Web Application Development:** A light and interactive web based program was developed to enable the model to reach the end users and especially the farmers and agricultural extension officers. The frontend was created with HTML, CSS, and Tailwind CSS which provides modern UI experience that is clean and possesses high levels of customizations. Tailwind CSS was chosen on the basis of its utility-first style, which enabled the creation of a very swift development and responsiveness that could be used across different devices with differently sized screen sizes. To develop the backend, FastAPI was selected because it is asynchronous and fast in execution and automatically generates an interactive API documentation (using Swagger UI). FastAPI also features the python type hints that helped make code much easier to read and saved time on debugging. This architecture provided an easy way of integrating with the trained machine learning models and implemented the real-time inference of images, with RESTful API endpoints.

- **Technology Decisions and Justification:** Other technologies had been considered during the project cycle. Both Flask and Django were evaluated as alternatives to deliver backend services but were ultimately disregarded in favour of FastAPI because of its better performance, current design and thorough support of asynchronous tasks- the latter being critical in keeping up with multiple simultaneous uploads and prediction. Local development environments were tried first, but cloud environments like Colab and Kaggle had more reliable compute resources and a faster turnaround of an experiment development loop. The necessity to have accessibility and platform independence was involved in the decision to build a web-based system rather than a native mobile application. By using a web interface, the system can be accessed by the oldest of smartphones and even kiosks with internet access, and now it can better be deployed to rural and under-resourced locations. Overall, the whole system that included development of the model, preprocessing and user interface was designed according to existing standards of software quality. The project will not only conform to the technical demands but also suit the wider objectives of sustainability, accessibility, and responsible deployment of AI in agriculture as it strives to be based on cloud platforms, modern frameworks, and open-source tools.

5.1.2 Hardware Standards

The Hardware infrastructure utilized in this project was specifically engineered such that available and consumer level Wares are used so as to encourage affordability, feasibility and deployment in actual practice. The system can also be made compatible with core engineering and communication standards, therefore facilitating optimum use and uniformity without involving advanced or dedicated AI hardware. The experimentation and development stage followed on HP Laptop (Model 16-e0xxx) with processor AMD Ryzen 7 5800H, NVIDIA GeForce RTX 3060 Graphics and 16GB DDR4 memory. This configuration was a strong balance of power and cost, it would serve to have enough computational resources to train the neural networks of deep learning algorithms without having to use cloud-based GPUs to do everything. In the performance of the system, it followed standard computing that associated with high speed in data processing such as the compatibility with DirectX graphics acceleration, and AMD 64-bits architecture. We used the iPhone 12 to take images with the aim of creating a dataset due to the high-resolution camera and reliability of image quality across all samples. This move can be classified as the compatibility to ISO 12233 as it determines the standards of the resolution of the electronic imaging system and as such the input information for use in the training must have high quality standards. Other factors include the fact that the phone passes the IEC 60065 safety standards of electronic equipment that contributes to a safe and secure data collection process, especially in the field.

To prevent clogging of network, the system supports the IEEE 802.11 (Wi-Fi) protocol so that updates or transfers of datasets can be carried out through the internet without any interruptions of communication due to failures in the network. The application was not being used on a mobile in this version, but a future portability was considered. Within that, the adherence to IEEE 1725, the safety standard that regulates the use of rechargeable batteries on the portable equipment, was considered as possible to apply to smartphones. A number of alternative courses of hardware were examined in the course of system design. The DSLR cameras were initially thought of because of better optics but were ruled out because of their expense, size, and practicality to deploy them in large fields. In a similar manner, low-power-consumption computers (single-board computers like Raspberry Pi and NVIDIA Jetson Nano) have been tested because of their power efficiency and modularity. Nevertheless, these platforms have presented a lot of complication in terms of setup and maintenance and also demanded more peripherals making them incompatible to the target consumers.

The last hardware approach was squared at compatibility, sustainability, and, the real world accessibility. The system is highly scalable with commercially available laptops and smartphones thus making the system affordable within the agricultural regions with limited resources. Each of the interfaces and components was chosen with a view toward maintainability, its future ability to fall within international standards of safety and interoperability.

5.1.3 Communication Standard

The designed system will include a hybrid communication strategy that will facilitate offline and online usage. The choice of the design increases flexibility in diverse deployment scenarios- especially in rural and unserved agrarian regions where internet connectivity is regularly unreliable or even lacking. On offline mode, the disease can be detected using local-based images without the use of networks, whereas in online mode, additional features including real-time synchronization of data including cloud storage, data synchronization, and updates can be enjoyed once network becomes available. In the online communication, the system makes use of standardized HTTP and HTTPS protocols. The implementation of the HTTPS protocol that introduces a secure level of LTLS/SSL encryption to HTTP provides a guarantee on safe data transport between the application and cloud-based services. This plays a critical role in securing sensitive user data like pictures, prediction results and feedback forms. Such universally accepted protocols make such solutions highly compatible with a modern web environment and also make the process of integration with various platforms much easier.

One RESTful API based architecture is implemented on the backend to control communications between the client and the server. Such architecture is stateless and modular, based on the communication model that enhances system flexibility and scalability. RESTful APIs facilitate future expansion of the system, be it to incorporate cloud-based analytics tools or a user dashboard or even add features that solicit and take action on feedback from farmers. Structured and resource-oriented data exchange is facilitated as well, and clarity and the performance of the communication processes are improved, as supported by the RESTful design. In the design process, alternative channels of communication were properly considered. Among the candidates was MQTT (Message Queuing Telemetry Transport, the low resource consumption high speed IoT application). Although MQTT is effective in delivery of data packets that are of small size, its weaknesses in large file delivery of media files like images of high resolution; thus, leaf images; as well as the customized deployment setup required, proved to be shortcomings of this use case.

FTP (File Transfer Protocol) was considered as well because of its simplicity and easiness of implementation. Nonetheless, it did not contain any encryption measures, which is quite a security risk, and it would not be applicable to use it in the context of transmitting confidential or personal data of the user in a modern application relying on AI. WebSockets, a protocol that provides full-duplex, real-time communication of data was also looked at. The functionality of WebSockets did technically allow delivering instant updates and two-way communication, but it resulted in the sometimes-wasted architectural complexity and resource requirements. Furthermore, it did not suit the circumstances of the system having to operate in areas with poor connectivity as well. Finally, the system embraces the HTTP/HTTPS together with RESTful APIs, as the favored communication model. It is a trade off between security, interoperability, integrative ease and performance. It also offers secure and trusted data transmission and maintains the capability to work without the connection to the internet. The REST architecture also allows the possibility of future scalability and modular building design because the system can be modified as the technological demands of the agricultural sector and the stakeholders increase.

5.2 Impact on Society, Environment and Sustainability

This section would assess the alignment of the proposed crop disease classification system to tomato, onion, and maize with the wider needs of the society, environmental responsibility and sustainability over the long term. Although the core way in which the system can be used is deemed to provide correct disease detection of the leaves, via the AI-driven model of image classification, there are much deeper implications of the system that surrounds its technical capabilities. Creation of a web-based, explainable AI model directly results in greater accessibility and inclusivity in agricultural technology. The low-resource/rural communities, especially smallholder farmers, do not have access to specialists in giving diagnostics. The system will allow these communities to make informed and timely decisions by giving them the power to classify the disease, in real time, using a web application, and in turn directly increase crop yield, food security, and economic resiliency.

Considering the sustainability side, the model was purposefully optimized to run, specifically, on the devices with consumer-grade Android, thus reducing the energy, as well as the computational overhead in general. Prediction, along with the capture of illustration, utilizes smartphones, which features scalability that does not demand extra hardware infrastructure. This is in line with sustainable development of technology, which makes it usable on a larger scale without further straining environments and economies. Additionally, the design of the system indicates the adherence to wholesome ethical principles of AI. It does not engage in data exploitation, safeguards the privacy of users and makes predictions transparent because the model design is explainable. Using fairness, accountability, and usability as the main priorities, the project advances responsible innovation in the field of agricultural AI. In short, the presented crop disease detection system fulfills the significant needs in society and it can be environmentally sustainable and ethically aligned with engineering. It serves as a case study on how technological progress can be used to not only address real life issues but make it more equitable, inclusive, and environmentally sustainable in nature.

5.2.1 Impact on Life

The suggested crop disease classification framework has high potential to empower small-scale and medium-scale farmers especially those that are in isolated and neglected regions. Agriculture-based economies such as the case of Bangladesh where farm economies rely heavily on the productivity of crops remains particularly critical in timely and proper identification of diseases that cause losses to crops of tomatoes, onion, and maize products.

The system prevents the development of plant diseases by allowing them to be detected at an early stage and thus high yields and economic sustainability among farmers are achieved. This proactive method mitigates the required use of expensive interventions and lessens the force of yield losses because of diseases, developing the social-economic resilience of agricultural societies. One of the aspects characteristic of the system is the application of explainable AI (XAI) techniques that are LIME (Local Interpretable Model-agnostic Explanations), SHAP (SHapley Additive exPlanations). The technologies provide open access to the model prediction results because they show visually what areas of the leaf image impacted which result in the model classification. This interpretability is likely to generate confidence on the part of the users particularly the farmers and agricultural advisors but it is also an excellent education tool as it enables users to gain more insight into what the plant disease indicators mean.

This system can be accessed by the web-based platform, which means that technology consumers will not need sophisticated devices or special web app to use the product to their advantage. Such a design decision enhances accessibility since farmers can upload pictures through internet-connected computers or community access locations that people can share. Its very basic and simplistic interface will allow an unguided novice person who does not have the technical skills required to effectively navigate the interface and use the system, without expert consulting or lab diagnostics. The platform fosters independence in decisions related to agriculture as it alleviates reliance on external sources of information and enables the users to have actionable insight. It contributes to the narrowing of the digital gap, as well as promotes the use of high-technology farming methods by deprived communities. In a wrap up, the web-based crop disease identification system can promote social inclusion, digital literacy in the agricultural setting, and solidify the premise of sustainable and self-reliant agriculture.

5.2.2 Impact on Society & Environment

The suggested software application of crop disease identification through the internet can be regarded as a massive input to the digitalization of the agricultural sector and specifically in the environment of the smallholder and resource-limited farming community. The system helps eliminate asymmetries in technology availability by providing a scalable, cost-efficient and easy to implement platform that covers rural and underserved areas. Its development enhances digital inclusion and equality, which allows consideration of the practice of precision agriculture without the precondition of specific hardware or an advanced level of supporting technical skills. This agricultural intelligence democratic landscaping furthers the wider message of increasing the urban rural digital divide and ensuring socioeconomic sustainability among impoverished farm communities.

To the society, the system has been an essential instrument in empowering the local stakeholders, most specifically the small and medium scale farmer through automatic early and reliable detection of diseases in tomato, onions, and maize crops. Diagnosis of the disease like rust, blight or fungal infections will occur early so that interventions can be employed sooner thus minimizing yields losses, assuring food security and improving the livelihoods of the farmers. To facilitate the explainability of the developed model, the system will be based on the human-centered AI with the explainable AI (XAI) approaches such as LIME, SHAP, Grad-CAM. These tools provide even users with graphical, intuitive descriptions of model output, which can enable users to better understand the model, trust model predictions, and learn and apply ideas about managing farms.

The system will help in promoting sustainable farming practices through promoting accuracy during application of chemical treatment in an environmental point of view. Instead of depending on indiscriminate or preventative measures of pesticides and fungicides farmers can identify afflicted plants or region depending on the feedback system. Such specimen-specific intervention minimizes chemical overflow, saves the integrity of the soil, and protects other ecosystems and water supplies in the surrounding areas against contamination. Being environmentally sound in terms of farming practices, the system promotes ecologically valid disease management practices.

In terms of sustainability, the system will be designed in such a way that it has to be run with less computational overhead. The optimization, lightweight deep learning models, namely AgriViTX and Vision Transformers running on a web platform limit the usage of continuous cloud-based processing and the energy requirements. This efficiency, in addition to making the system more responsive also leads to reduced carbon foot print. Moreover, the system will support intermittent connectivity and, therefore, will be robust and accessible in low-resource environments to be able to effectively run over the long term.

5.2.3 Ethical Aspects

Ethics was used in every step of developing the system. Every data on image were only gathered upon gaining the consent of the landowner of commercial tomato, maize, and onion farms. No personally identifiable information was stored during the acquisition process; photos of the leaves were obtained using regular digital cameras and the recording of it was anonymized before storage. To protect against the situations of bias, the model was thoroughly tested in terms of being fair at different field conditions, ranging from various illumination, background heterogeneity, and leaf orientations. This validation reduces the possibility of inaccurate diagnoses and facilitates consistency in reliability irrespective of the manner in which the end users present their images. The explainable AI techniques (LIME, SHAP, attention heatmaps) ensure transparency in decision making. In place of an incomprehensible black box, the system creates graphical highlights of which parts of an image influenced which prediction. The feature gives the user the ability to check, inspect, and comprehend the reasoning behind the model, which upholds the concepts of autonomy, accountability, and non-maleficence. Data governance was developed in a way that it did not infringe on the rights of users. No photos or feedbacks are saved or shared without clear permission of a user. Confidentiality and data integrity are guaranteed by encryption in all the communication between the network including synchronization, especially with cloud-based applications, which is carried out using HTTPS. These will make the project relevant to the best practices in the field of digital privacy and ethical AI implementation

5.2.4 Sustainability Plan

A multi-dimensional plan of sustainability has been put in place to ensure that once the objectives of the system are achieved; the system sustains itself appropriately on all fronts (four dimensions):

- **Technical Sustainability:** Core AgriViTX and Vision Transformer models are designed to have the minimal possible number of parameters that perform well on both the inference of common web servers and lower-end client devices. Usage of Python and specifically TensorFlow/Keras, with well-known machine-learning tooling, will make it easy to maintain and improve in the future. Its modular architecture also allows that it can be updated modularly by including new categories of diseases or by perfecting some of the components of the model, without going through a total redesign of the platform.

- **Economic Sustainability:** This solution also reduces implications on implementation costs and operating costs because it uses only open-source software, cloud-based platforms, where the solutions can be accessed (e.g., Google Colab, Kaggle) and commodity hardware (running on standard laptops and connected devices). The offline features also cut down on constant expenses by minimizing their requirements in constant data transfer or cloud-based servers. The model can be deployed on a large scale in areas that have a low budget.

- **Environmental Sustainability:** It encourages on-site low-energy processing, by inference on the browser, or on resource-light server instances. With accurate disease detection, farmers can use the agrochemicals only when needed and the amount of chemical runoff will be considerably reduced and save both soil condition and adjacent ecosystems. Less overall carbon footprint is also attributed to reduced data-centers as they run continuously in cloud computing.

- **Sustainability of community and knowledge:** Explainable AI elements convert the platform into a learning aid RAEEs and farmers may utilize in interpreting model results and using best practices when dealing with diseases. Open-source publication plans of future releases will also lead to the development of a community that helps with improvements to the current platforms, the localization of the software to other crops and the sustainability of the project in the long term.

5.3 Project Management and Financial Analysis

This section describes the business model of financial planning, the level of resources that is likely to be incurred, the patterns of the allocation of resources and the revenue platforms that might arise with the system. Although the presented work was not sponsored commercially and developed independently, a cost model that reflects the real cost must also be produced in order to assess future scalability, operation deployment and market feasibility. In order to cover possible implementation scenarios, two scenarios of budget projections are outlined- a comparison of the low-cost, open-source/self-supported development cost model in this study with the estimated cost of a commercial/enterprise-scale implementation. It is through this comparative break down where the economic sense of the system is seen under the two aspects such as the grassroots level compared to industrial level.

5.3.1 Project Planning and Task Management

This research was also successfully implemented because of the successful management of the project. Since the study was conducted independently, all the tasks performed within the study were well-structured, planned and supervised in order to complete in time preserving the scientific rigor and reproducibility. It is important to note that the project was structured in well-understood, purpose-oriented phases, which provided the project with the possibility of efficient movement and responsiveness to the subsequent revelation or failure. The research and development process were based on a milestone approach that involved an extensive background research and reports, training and testing of the model, incorporation of explainable AI (XAI) methods, and finally deployment of the system design as a lightweight, web-based interface. The phases were mutually supportive, and the results of the previous phases were explicitly included in the decision and the refinements of the upcoming ones. Iterative experimentation and model improvement were provided with the use of Agile methodologies. To correct at every stage feedback loop was instituted in place to determine the performance. The Google Colab Pro was also used to ensure that the required GPU-powered infrastructure was employed in the development of the model through Python-based deep learning libraries including TensorFlow and Keras. The last stage of deployment was devoted to the development of the responsive web application as a means of delivering the trained and validated model to the end user. Fast-api was leveraged to prototype and deploy the interface that would provide the user with an interactive disease prediction interface and visual explanations (e.g. Grad-CAM visualization overlays). The web-based deployment was actually preferable due to its wider access by removing the possibility of users to install the application and the need to use high-performance machines. Explicit consideration was put on how to optimize on low-bandwidth connections and making it compatible with various browsers and devices.

Table 5.1: Project Timeline

Phase	**Start Date**	**End Date**	**Duration**
Initial Documentation (Intro, Literature Review, Gap Analysis)	April 7, 2025	April 16, 2025	2 weeks
Data Collection & Preprocessing	April 18, 2025	April 20, 2025	1 weeks
Methodology Design	May 6, 2025	May 10, 2025	1 weeks
Experimentation with Transfer Learning Models	June 15, 2025	June 28, 2025	3 weeks
Development of AgriViTX Model	July 1, 2025	July 20, 2025	3 weeks
XAI Integration & Model Evaluation	July 21, 2025	July 30, 2025	2 weeks
Web Application Deployment	August 11, 2025	August 20, 2025	1.5 weeks
Final Documentation & Printing	September 08, 2025	September 08, 2025	1 day

Digital tools such as Google Calendar and Trello helped plan a project and monitor its progress, and version control and reproducibility of code were possible with the help of GitHub. The last stage of the implementation process was concerned with the transition of the tested model of AgrViTX, validated in the research, into a convenient and user-friendly online interface. Particular note was given to the design of the user interface, speed of inference, ability to operate online and compatibility across devices, as most researchers and other workers operating in crop-growing areas may have limited technical expertise.

5.3.2 Financial Analysis

The financial analysis makes the evaluation based on the actual costs involved in the development of the crop disease detection system (within tomato, onion, and maize) as well as the potential future costs that might be involved in the mass deployment or commercialization of the system. Since this study was pursued independently on an academic basis, a cost-effective approach was taken through the use of personal computing resources, open-source software libraries, and cheap cloud computing services like Google Colab Pro on GPUs to get GPU-accelerated training. Such cost-effective choices helped to perform proper experimenting with models and web-based deployment without any worsening of quality or performance. A hypothetical situation of a commercial budget was also bearable to promote scalability in the future. This alternative projection considers enterprise level infrastructure, paid infrastructure, and professional level development infrastructure--providing a more accurate view on production implementation, especially in areas that need a wider level of technology expansion or adoption in institutions.

Table 5.2: Actual Research Budget (Self-Supported, Research-Based)

Component	**Estimated Cost (BDT)**	**Remarks**
Internet and Cloud Resources	2,000/month × 4 = 8,000	Kaggle for model training and testing
Software Tools	0	Python, Keras, OpenCV, and other libraries are open source
Image Preprocessing Tool	0	Microsoft PowerToys (free image resizer)
Thesis Printing and Binding	3,000	For hardcopy submission
Miscellaneous (USB, Cables, etc.)	2,000	Small accessories
Total Cost	13,000 BDT	

Table 5.3: Alternate Budget (Enterprise-Scale Deployment)

Component	**Estimated Cost (BDT)**	**Remarks**
Commercial Smartphone (for app testing)	25,000	Mid-range Android device for target testing
Cloud Compute Credits (GCP or AWS)	15,000	Model training, fine-tuning on high-performance GPUs

Web Developer License	2,000	For publishing on Web
UI/UX Development (Freelance/UI Expert)	10,000	Professional UI design
Marketing & Awareness Campaigns	15,000	Promotion among farmers and crop estates
Maintenance, Updates & Customer Support	10,000/year	Optional support and versioning
Total Estimated Budget	77,000 BDT	

5.4 Complex Engineering Problem

This section underscores the engineering challenge that is explained in this thesis as being complex. The work demanded an interdisciplinary approach to integrate knowledge in computer vision, transfer learning, deep learning, and explainable AI (XAI) with agriculture knowledge to effectively design an accurate, interpretable classification system of crop diseases (in tomato, onion, and maize crops) that is compatible with the web. The project did not only require professional expertise but also focused on pragmatic design factors like the limitation of resources, the efficiency of a model, and its access by final users in the rural or resource-limited contexts.

The most important factors considered were the robustness to different lighting and background conditions, the ability to adapt the system to low-bandwidth conditions, and the interpretability, using these XAI techniques. Conversations related to scalability, future extensibility (i.e. adding new crops types or diseases), and fair access to the users played a marquee role in architectural and implementation design choices. The subsequent subsections explain how the work represents an example of finding solutions to challenging engineering problems in addition to being congruent with established known profiles of engineering knowledge and practices.

5.4.1 Complex Problem Solving

To overcome the various dimensions of complexity that face this research, the project used an array of engineering tools, namely, algorithm choice, model architecture integration, optimization to lightweighted inference, and deployment of real time web. The work fits the Engineering Problem (EP) framework with the table below:

Table 5.4: Mapping with complex problem solving.

EP1- Depth of Disciplinary Knowledge	EP2- Conflicting Requirements	EP3- Rigor and Depth of Analysis	EP4- Domain Familiarity	EP5- Extent of Applicable Codes	EP6- Extent Of Stake-holder Involvement	EP7- System Interdependence
✓	✓	✓	✓			✓

Justifications

- **EP1 – Depth of Disciplinary Knowledge:** The research needed to be based on knowledge of convolutional neural networks, Vision Transformer architectures, transfer learning, advanced data augmentation (using OpenCV/PIL) and explainability methods (LIME, SHAP, attention heatmaps). Besides, the final deployment required the competency with web development (HTML, CSS, Tailwind, FastAPI).

- **EP2 – Conflicting Requirements:** The system traded off competing goals: this is a high-value classification system: high classification accuracy on tomato, onion and maize diseases, a small model footprint and the ability to do offline classification in settings where web access is hard.

- **EP3 – Rigor and Depth of Analysis:** A detailed experimental pipeline that compared the six different pretrained CNNs and one ViT model on various metrics (accuracy, precision, recall, F1-score). The model that performed best was further optimized so that it has a lightweight AgriViTX and is energy efficient in inference.

- **EP4 – Domain Familiarity:** The previous experience of conducting a study in agricultural image analysis was useful to define the way to address the problem of class imbalance, variable lighting, and background clutter. This field of knowledge informed the architecture in preprocessing, augmentation and evaluation procedures.

- **EP7 – System Interdependence:** The architecture is modular and there is easy integration of varying modular backbones, XAI modules, and web frameworks. The core of AgriViTX is light and may support other types of crops or sets of diseases without extensive redesign.

Mapping with Knowledge Profile for EP1

Table 5.5: Mapping with knowledge Profile.

K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K8 Research Literature
✓		✓	✓	✓

Justifications

- **K3 – Engineering Fundamentals:** Examples of the core concepts used are classification algorithms, optimization methods, image preprocessing (resizing, normalization), and evaluation of performance with standard measurements.

- **K5 – Engineering Design:** The workflow that was used to design the system includes the choice of appropriate CNN and ViT models, the development of a customized AgriViTX framework, combining XAI to improve the interpretability of the results, and deployment in the form of a responsive web app based on FastAPI and Tailwind CSS.

- **K6 – Engineering Practice:** After collecting data, preprocessing the data, training and explaining the model, the research followed an established process of work processes, the best practices include version control (GitHub), code readability (PEP 8), and model repeatability (Colab/Kaggle).

- **K8 – Research Literature:** In-depth literature review has found areas of multicrop disease detection, lightweight models deployment, and XAI incorporation lacking in the topics. Key ideas during the current research were informative in model selection, data strategies, and transparency methods as used in this thesis.

5.4.2 Engineering Activities

The project lifecycle consisted of a set of complex engineering tasks- data acquisition, model development and training, performance and explainability visualization, web-based deployment. Such stages are the illustration of the technical issues of agricultural AI-based solution delivery in real-life conditions.

Table 5.6: Mapping with complex engineering activities.

EA1 Range of sources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and the environment	EA5 Familiarity
✓		✓	✓	✓

Justifications

- **EA1 – Diversity of Tools and Resources:** Multiple platforms and technologies were utilized in the project: smartphone cameras to collect images in the field; Python, NumPy, and OpenCV to preprocess the data; TensorFlow/Keras to train CNN and Vision Transformer models; LIME and SHAP to interpret the models; and an FastAPI with Tailwind CSS to deploy a web app. Training of the models was carried out in Google Colab and Kaggle GPU-accelerated.

- **EA3 – Degree of Innovation:** This work presents an end-to-end pipeline - beginning with multi-crop field data collection to a lightweight explainable deep learning model deployed via a responsive web interface. In constrast to most other high-performance systems, which are also resource-intensive, this system offers a tradeoff between accuracy, interpretability, and in a resource-limited setting.

- **EA4 – Societal and Environmental Impact:** The platform also creates opportunities to prevent the emergence of diseases in tomato, onion, and maize through early diagnosis, allowing farmers to implement precise interventions and thus minimizing the use of agrochemicals, decreases erosion of soil and water purity, and increases harvest. Such results are providing direct livelihood and environmental sustainability results to the community.

- **EA5 – Familiarity with Methods and Technologies:** All used methods and tools, including convolutional architectures, Vision Transformers, XAI techniques, and Python-based web frameworks, were chosen because of the previous experience of the research team. This expertise in this domain helped in quick prototyping, extensive experimentation, and iteration of the system.

5.5 Summary

This chapter gives a clear overview of the fundamental engineering principles and intricate design factors that will be used in coming up with a disease detection system of crops that is being proposed. The project was an interdisciplinary field requiring the combination of computer vision and deep learning (CNN and Vision Transformers), explainable AI (XAI) and lightweight deployment that can be used in web settings. In ranging all the way through gathering crop leaf images under real-world conditions to constructing and implementing a functional application, both of them were pegged on the real-world farming applications. It was worked out to develop a custom lightweight CNN (AgriViTX) in order to maintain high accuracy and minimize computational footprint, and thus the solution fits number-constrained web platforms perfectly. The project solved some of the major issues in society by early and accessible detection of disease in tomato, onion, and maize-farmers i.e., small holder farmers and food security. Ethical and environmental considerations have been built into the device as well, with on-device processing and ethically-minded design-time implementation, and energy-efficient implementation. Standard frameworks were used to plot all complex engineering problem solving activities and outcomes showed that they had met the key attributes of depth of knowledge, conflicting requirements as well as interdependence. Moreover, the piece of work met the engineering activity requirements that entail innovation, societal value, and the utilization of common tools and techniques. The engineering principles, engineering design aspects, and perspectives of the ongoing studies were always utilized during the development process to direct the system integration and system improvement.

Chapter 6

Conclusion

The main findings of the research, the limitations encountered during the implementation process and suggestions on the further development are discussed in this chapter. The achieved overall results in the model exploring earlier studies, employing explainable AI methods, and the following model implementation into an actual real-time web application underline the practical usefulness of the presented CNN and ViT-based system recognizing plant diseases.

6.1 Summary

This paper focused on designing an effective, predictable and interpretable deep learning model of automatic diagnosis of multiple crop leaf conditions on tomato, onion and maize plants. Here, the main contribution is the development of a deployable AI system that combines the capabilities of powerful deep learning model with the explainability of artificial intelligence (XAI), which is optimized to be deployed in a real-time environment as well as web-friendly in agriculture applications. To test different deep learning approaches, a diverse crop leaf dataset composed of many images was used to train a number of standard CNN architecture on which the network was fine-tuned. These were the ResNet152V2, DenseNet201, InceptionV3, InceptionResNetV2, Xception and ResNet101. In these, ResNet152V2 achieved the best overall performance as it was tested to return 96.56% accuracy and DenseNet201 was testing to give 95.42% accuracy. These models had high generalization with seen and unseen data and good classification in the three crops. ResNet152V2 and DenseNet201 yielded the highest test accuracy of above 95 percent in the tomato dataset proving their effectiveness in the prediction of prevalent tomato diseases. Superior performance of these models was also emphasized in the onion dataset analysis, and ResNet152V2 reached perfect F1-scores of 0.99 in both Virosis Disease and Healthy Leaves classes or almost identical high performance on all classes in the DenseNet201 model, which would enable them to be deployed in real-world crop monitoring tools. The model had the same high precision, recall, and F1-scores on all 7 disease and healthy classes in the maize dataset. F1-scores were between 0.97-0.99, particularly on important diseases such as Rust, Curvularia and Helminthosporiosis meaning that the model is not only precise, but also resilient to multiple types of plants and visual manifestations of diseases. Although such deep CNNs performed incredibly well, their associated computational costs and number of parameters are exorbitant and unsuitable to be used in resource sparse setups such as in rural farms or smartphones. In order to handle this difficulty, lightweight models optimization was also a concern in the project. One of the stated projects entailed the creation of a custom lightweight CNN that would shrink the model to offer competitive results but equally provide highly reduced size so that the scope of the AI could be accessed with ease by the low-resource individuals and organizations (such as farmers and field workers). The system was designed with XAI tools including SHAP, LIME, and Grad-CAM to facilitate the desire to bring interpretability and transparency to decision-making. These tools offered visual and feature-based explanations on the predictions to the model that is critical towards building trust in their users particularly in the agricultural sphere wherein misdiagnosis can translate to huge losses. The use of XAI also helped make sure that the system did make accurate predictions and users and agronomists can comprehend the reasoning behind specific classifications that were made. The system was designed with XAI tools including SHAP, LIME, and Grad-CAM to facilitate the desire to bring interpretability and transparency to decision-making. These tools offered visual and feature-based explanations on the predictions to the model that is critical towards building trust in their users particularly in the agricultural sphere wherein misdiagnosis can translate to huge losses. The use of XAI also helped make sure that the system did make accurate predictions and users and agronomists can comprehend the reasoning behind specific classifications that were made.

6.2 Limitation

Even though the proposed system has managed to deliver some positive outcomes, there still are some limitations to this research that must be improved in the next studies.

- First, the datasets are not taken by direct field sampling, but from publicly available repositories. Although these datasets gave adequate images to experiment on, they might not entirely reflect the broad diversity of the actual agricultural landscape, such as the geography, climate, and farming methods. This would limit the generalization capability of the model in different farming settings.

- Second, the maize, tomato, and onion datasets used in the experiment are relatively small against the global spread of crop diseases, although they are not insufficient. Specifically, the dataset of onions was small in terms of size and this could have influenced the model to produce the entire spectrum of variations of the diseases. It will be beneficial to increase the sample sizes with samples covering other areas, disease progression stages, and environmental factors in order to reduce bias and enhance the strength of the model.

- Third, the proposed model, which is lightweight and efficient, might not perform well in more complex settings like overlapping leaves, mixed disease manifestations, or cluttered backdrops. In such cases advanced or hybrid architectures may potentially enhance accuracy but also may impose increased computational costs, making real time implementation.

- Fourth, despite the use of Explainable AI (XAI) methods (SHAP, and Pseudo Segmentation) during the experimentation process, they were not incorporated into the created web-based application. This restricts the ability of end-users e.g. farmers and agricultural advisors to visually understand the reasoning behind the model, which is a requirement in establishing trust and transparency in decision making.

- Lastly, even though a functional web-based application was created and tried, a lack of a mobile application makes it less accessible to farmers located in rural environments. The system can be made more realistic, convenient, and effective in terms of simple agricultural decision-making directly in the field by having a special mobile application that would enable farmers to take leaf pictures and process them to the best of their ability.

6.3 Future Work

Based on the results of this investigation, various opportunities can be pursued to make the proposed crop disease detection system more reliable, scaleable, and usable. One of the main ways is the expansion and diversification of the datasets. Although this work used publicly available repositories, the datasets are small in geographic scope and seasonal variety, and the onion dataset in the specific case is limited. The model would benefit by gathering more images across geographical locations, seasons, and disease progression stages to enhance its generalizability and resilience in real-world agricultural systems. Another valuable development is the inclusion of Explainable AI (XAI) within the deployed applications themselves. Granted, when the model was evaluated, tools like Grad-CAM, LIME, and SHAP were used, but they are not integrated into the web solution yet. The XAI techniques can be embedded into the system in ways that are lightweight and mobile to enable farmers and agricultural advisors visually interpret predictions, which would boost acceptance and use of the system. Future studies can also aim at making Vision Transformer (ViT) model even more efficient. Techniques like model pruning, quantization and knowledge distillation can further minimize the computation time and memory required, making the system more appropriate to run on low-power mobile and edge devices deployed in rural areas. Also, it would be beneficial to expand the framework to an exclusive mobile application that would equip farmers with the ability to detect diseases on-field, making its use more viable and reachable than when a web platform is the only tool. Lastly, the system can be expanded in two directions namely: (i) to support more crops and diseases to provide a multi-crop solution and (ii) to go beyond mere classification to actually estimate the severity of the diseases or at least monitor disease development over a period of time. These improvements, potentially aided by multimodal information (e.g., leaf images and the environmental variables of temperature, humidity, and soil conditions), would allow more effective and proactive crop health management.

References

[1] Aboelenin, S., Elbasheer, F. A., Eltoukhy, M. M., El-Hady, W. M., & Hosny, K. M. (2025). A hybrid Framework for plant leaf disease detection and classification using convolutional neural networks and vision transformer. Complex & Intelligent Systems, 11(2), 142.

[2] Ahila Priyadharshini, R., Arivazhagan, S., Arun, M., & Mirnalini, A. (2019). Maize leaf disease classification using deep convolutional neural networks. Neural Computing and Applications, 31, 8887-8895.

[3] Albahli, S., & Masood, M. (2022). Efficient attention-based CNN network (EANet) for multi-class maize crop disease classification. Frontiers in Plant Science, 13, 1003152.

[4] Amin, H., Darwish, A., Hassanien, A. E., & Soliman, M. (2022). End-to-end deep learning model for corn leaf disease classification. IEEE Access, 10, 31103-31115.

[5] Anandhakrishnan, T., & Jaisakthi, S. M. (2022). Deep Convolutional Neural Networks for image-based tomato leaf disease detection. Sustainable Chemistry and Pharmacy, 30, 100793. https://doi.org/10.1016/j.scp.2022.100793

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

[7] Ashwini, C., & Sellam, V. (2024). An optimal model for identification and classification of corn leaf disease using hybrid 3D-CNN and LSTM. Biomedical Signal Processing and Control, 92, 106089.

[8] Bhagat, M., Kumar, D., & Kumar, S. (2024). Optimized transfer learning approach for leaf disease classification in smart agriculture. Multimedia Tools and Applications, 83(20), 58103-58123.

[9] Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2021). Deep Learning for Tomato Diseases: Classification and Diagnosis Using Leaf Images. Computers and Electronics in Agriculture, 152, 139–146. https://doi.org/10.1016/j.compag.2018.07.031

[10] Elmasry, A., Abdullah, W., Kang, B. G., & Nam, Y. (2024). A novel hybrid approach based on CNN for corn diseases detection. Optimization in Agriculture, 1, 94-104.

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

[12] Gulzar, Y., Hamid, Y., Soomro, A. B., Alwan, A. A., & Journaux, L. (2020). A convolution neural network-based seed classification system. *Symmetry*, *12*(12), 2018.

[13] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[14] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[15] Kavukcuoglu, K., Sermanet, P., Boureau, Y. L., Gregor, K., Mathieu, M., & LeCun, Y. (2010). Learning convolutional feature hierarchies for visual recognition. In Advances in Neural Information Processing Systems, 1090–1098.

[16] Kayaa, A., Kecelia, A. S., Catalb, C., Yalica, H. Y., Temucina, H., & Tekinerdoganb,

B. (2019). Analysis of transfer learning for deep neural network-based plant classification models. Computers and Electronics in Agriculture, 158, 20–29. https://doi.org/10.1016/j.compag.2019.01.041

[17]Chowdhury, M. E., Rahman, T., Khandakar, A., Ayari, M. A., Khan, A. U., Khan, M. S., ... & Ali, S. H. M. (2021). Automatic and reliable leaf disease detection using deep learning techniques. AgriEngineering, 3(2), 294-312.

[18]Yalcin, H., & Razavi, S. (2016, July). Plant classification using convolutional neural networks. In *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)* (pp. 1-5). IEEE.Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science, 7, 1419. https://doi.org/10.3389/fpls.2016.01419

[19]Thangaraj, R., Anandamurugan, S., & Kaliappan, V. K. (2021). Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. Journal of Plant Diseases and Protection, 128(1), 73–86. https://doi.org/10.1007/s41348-020-00386-4

[20]Kaya, A., Keceli, A. S., Catal, C., Yalic, H. Y., Temucin, H., & Tekinerdogan, B. (2019). Analysis of transfer learning for deep neural network based plant classification models. *Computers and electronics in agriculture*, *158*, 20-29.

[21]Negi, A., Kumar, K., & Chauhan, P. (2021). Deep neural network-based multi-class image classification for plant diseases. *Agricultural informatics: automation using the IoT and machine learning*, 117-129.

[22]Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science, 7, 1419. https://doi.org/10.3389/fpls.2016.01419

[23]Khan, Olayiwola, & Kaliappan, V. K. (2021). Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. Journal of Plant Diseases and Protection, 128(1), 73–86. https://doi.org/10.1007/s41348-020-00386-4

[24]Trivedi, N. K., Gautam, V., Anand, A., Aljahdali, H. M., Villar, S. G., Anand, D., ... & Kadry, S. (2021). Early detection and classification of tomato leaf disease using high-performance deep neural network. Sensors, 21(23), 7987. https://doi.org/10.3390/s21237987

[25]Panigrahi, X., Zeng, Q., Luo, X., & Chen, L. (2022). FCAU-Net for the Semantic Segmentation of Fine-Resolution Remotely Sensed Images. Remote Sensing, 14(1), 215.

[26]Olayiwola, Z., Geng, Y., Wang, C., Xue, Y., Sun, D., Lou, Z., ... & Quan, L. (2024). InstaCropNet: an efficient Unet-based architecture for precise crop row detection in agricultural applications. Artificial Intelligence in Agriculture, 12, 85-96.

[27]Wenxia, S., Liu, Y., Wang, J., & Song, H. (2021). A decade survey of transfer learning (2010–2020). IEEE Transactions on Artificial Intelligence, 1(2), 151-166.

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

[29]Thangaraj, R., Anandamurugan, S., & Kaliappan, V. K. (2021). Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. Journal of Plant Diseases and Protection, 128(1), 73–86. https://doi.org/10.1007/s41348-020-00386-4

[30]Kayaa, A., Kecelia, A. S., Catalb, C., Yalica, H. Y., Temucina, H., & Tekinerdoganb,

B. (2019). Analysis of transfer learning for deep neural network-based plant classification models. Computers and Electronics in Agriculture, 158, 20–29. https://doi.org/10.1016/j.compag.2019.01.041

[31]López-Martínez, M. J., Díaz-Flórez, G., Villagrana-Barraza, S., Castañeda-Miranda, C. L., Solís-Sánchez, L. O., Ortíz-Esquivel, D. I., ... & Olvera-Olvera, C. A. (2024).Pattern Classification of an Onion Crop (Allium Cepa) Field Using Convolutional Neural Network Models. Agronomy, 14(6), 1206.

[32]Khan, S. P., Shohel, D. P., & Salathé, M. (2016). Deep Learning for Plant Disease Detection. Frontiers in Plant Science, 7, 1419.

[33]Attallah, O. (2023). Tomato leaf disease classification via compact convolutional neural networks with transfer learning and feature selection. Horticulturae, 9(2), 149.

[34]Amin, H., Darwish, A., Hassanien, A. E., & Soliman, M. (2022). End-to-end deep learning model for corn leaf disease classification. IEEE Access, 10, 31103-31115.

[35]Sun, J. (2016), He, K., Zhang, X., Ren, S.,. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[36]Nguyen, T. H., Nguyen, T. N., & Ngo, B. V. (2022). A VGG-19 model with transfer learning and image segmentation for classification of tomato leaf disease. AgriEngineering, 4(4), 871-887.

[37]Guo, N. K., Gautam, V., ... & Kadry, S. (2021). Early detection of multi-crop leaf disease using deep neural network.https://doi.org/10.3390/s21237987

[38]Shandilya, G., Gupta, S., Mohamed, H. G., Bharany, S., Rehman, A. U., & Hussen, S. (2025). Enhanced Maize Leaf Disease Detection and Classification Using an Integrated CNN-ViT Model. Food Science & Nutrition, 13(7), e70513.

[39]Barman, U., Sarma, P., Rahman, M., Deka, V., Lahkar, S., Sharma, V., & Saikia, M. J. (2024). Vit-SmartAgri: vision transformer and smartphone-based plant disease detection for smart agriculture. Agronomy, 14(2), 327.

[40]El Sakka, M., Mothe, J., & Ivanovici, M. (2024). Images and CNN applications in smart agriculture. European Journal of Remote Sensing, 57(1), 2352386.

[41]Fu, X., Ma, Q., Yang, F., Zhang, C., Zhao, X., Chang, F., & Han, L. (2024). Crop pest image recognition based on the improved ViT method. Information Processing in Agriculture, 11(2), 249-259.

[42]Ouamane, A., Chouchane, A., Himeur, Y., Miniaoui, S., Atalla, S., Mansoor, W., & Al-Ahmad, H. (2025). Optimized vision transformers for superior plant disease detection. IEEE Access.

[43]Parez, S., Dilshad, N., Alghamdi, N. S., Alanazi, T. M., & Lee, J. W. (2023). Visual intelligence in precision agriculture: Exploring plant disease detection via efficient vision transformers. Sensors, 23(15), 6949.

[44]Nishankar, S., Pavindran, V., Mithuran, T., Nimishan, S., Thuseethan, S., & Sebastian, Y. (2025). ViT-RoT: Vision Transformer-Based Robust Framework for Tomato Leaf Disease Recognition. AgriEngineering, 7(6), 185.

[45]Ali, M., Salma, M., Haji, M. E., & Jamal, B. (2025). Plant disease detection using vision transformers. Int J Electr Comput Eng (IJECE), 15(2), 2334-44.

[46]Islam, M. M., Adil, M. A. A., Talukder, M. A., Ahamed, M. K. U., Uddin, M. A., Hasan, M. K., ... & Debnath, S. K. (2023). DeepCrop: Deep learning-based crop disease prediction with web application. *Journal of Agriculture and Food Research*, *14*, 100764.

[47]Aasha Nandhini, S., Hemalatha, R., Radha, S., & Indumathi, K. (2018). Web enabled plant disease detection system for agricultural applications using WMSN. *Wireless Personal Communications*, *102*(2), 725-740.

[48]Kheir, A. M., Koubaa, A., Kolluru, V., Mungara, S., & Feike, T. (2025). Smart Plant Disease Diagnosis Using Multiple Deep Learning and Web Application Integration. *Journal of Agriculture and Food Research*, 101948.

[49]Manasa Sakka, M., Mothe, J., & Ivanovici, M. (2024). Images and CNN applications in smart agriculture. European Journal of Remote Sensing, 57(1), 2352386.

[50]Obadaarachchi, M, D., & Kumar, S. (2023). Optimized transfer learning approach for leaf disease classification in smart agriculture. Multimedia Tools and Applications, 83(20), 58103-58123.

[51]Tembhurne, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161, 272-279.

[52]Appiah, Obed; Hackman, Kwame Oppong ; Diallo, Belko Abdoul Aziz; Ogunjobi, Kehinde O; Ouedraogo, Valentin; Bebe, Momo; SON, Diakalia (2024), "TOM2024", Mendeley Data, V1, doi: 10.17632/3d4yg89rtr.1

[53]Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, *53*(8), 5455-5516.

[54]Singh, T., & Vishwakarma, D. K. (2021). A deeply coupled ConvNet for human activity recognition using dynamic and RGB images. *Neural Computing and Applications*, *33*(1), 469-485.

[55]GC, S., Saidul Md, B., Zhang, Y., Reed, D., Ahsan, M., Berg, E., & Sun, X. (2021). Using deep learning neural network in artificial intelligence technology to classify beef cuts. *Frontiers in Sensors*, *2*, 654357.

[56]Akhtar, M. J., Mahum, R., Butt, F. S., Amin, R., El-Sherbeeny, A. M., Lee, S. M., & Shaikh, S. (2022). A robust framework for object detection in a traffic surveillance system. *Electronics*, *11*(21), 3425.

[57]Ahad, M. T., Song, B., & Li, Y. (2024). A Comparison of Convolutional Neural Network, Transfer Learning and Ensemble Technique for Brain Tumour Detection of Classification. *Transfer Learning and Ensemble Technique for Brain Tumour Detection of Classification*.

[58]Krithika, M., & Gayathri, A. (2024, December). Beyond Depth: Evaluating ResNet-50 and ResNet-152 Performance on Cifar-10 Dataset. In *2024 International Conference on Sustainable Communication Networks and Application (ICSCNA)* (pp. 973-977). IEEE.

[59]Mehmood, A. (2021). Efficient anomaly detection in crowd videos using pre-trained 2D convolutional neural networks. *IEEE access*, *9*, 138283-138295.

[60]Assaduzzaman, M., Bishshash, P., Nirob, M. A. S., Al Marouf, A., Rokne, J. G., & Alhajj, R. (2025). XSE-TomatoNet: An explainable AI based tomato leaf disease classification method using EfficientNetB0 with squeeze-and-excitation blocks and multi-scale feature fusion. MethodsX, 14, 103159.

[61]Afroj, M., Jahan, I., Islam, M. R., & Islam, M. P. (2025). Detection of Corn Leaf Diseases Using Deep CNN with SHAP-Based Explainability for Enhanced Agriculture. In *Machine Vision in Plant Leaf Disease Detection for Sustainable Agriculture* (pp. 81-90). Singapore: Springer Nature Singapore.

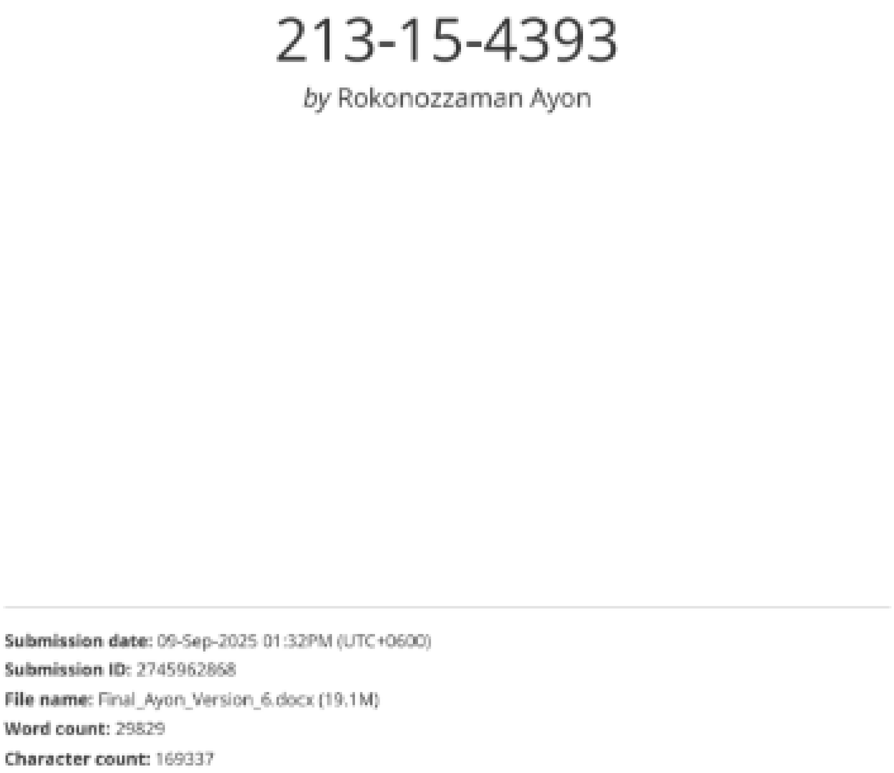
[62]Bhandari, M., Shahi, T. B., Neupane, A., & Walsh, K. B. (2023). Botanicx-ai:

Identification of tomato leaf diseases using an explanation-driven deep-learning model. Journal of Imaging, 9(2), 53.

[63]Arora, M., Santra, A., Pathak, D., Agrawal, M., Chopra, S., & Vachhani, H. (2025). Interpretable Deep Learning for Sustainable Agriculture: CNN and LIME-Based Plant Disease Diagnosis. *International Journal of Environmental Sciences*, *11*(9s), 1016-1030.

[64]You, J., Liu, W., & Lee, J. (2020). A DNN-based semantic segmentation for detecting weed and crop. *Computers and Electronics in Agriculture*, *178*, 105750.

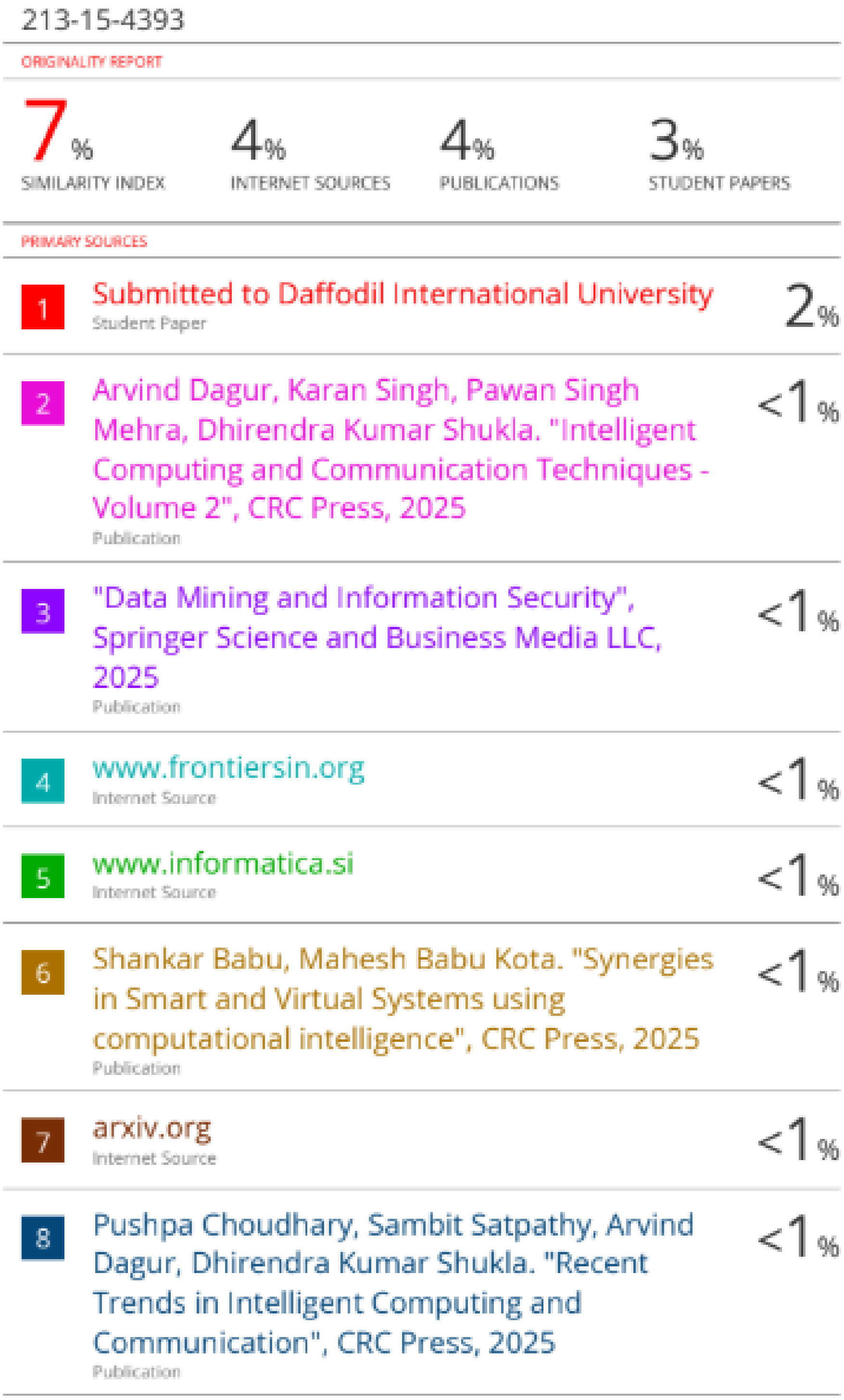
[65]Ciarfuglia, T. A., Motoi, I. M., Saraceni, L., & Nardi, D. (2022). Pseudo-label generation for agricultural robotics applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1686-1694).



213-15-4393

by Rokonozzaman Ayon

Submission date: 09-Sep-2025 01:32PM (UTC+0600)
Submission ID: 2745962868
File name: Final_Ayon_Version_6.docx (19.1M)
Word count: 29829
Character count: 169337



213-15-4393

ORIGINALITY REPORT

7%	4%	4%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	2%
2	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Intelligent Computing and Communication Techniques - Volume 2", CRC Press, 2025 Publication	<1%
3	"Data Mining and Information Security", Springer Science and Business Media LLC, 2025 Publication	<1%
4	www.frontiersin.org Internet Source	<1%
5	www.informatica.si Internet Source	<1%
6	Shankar Babu, Mahesh Babu Kota. "Synergies in Smart and Virtual Systems using computational intelligence", CRC Press, 2025 Publication	<1%
7	arxiv.org Internet Source	<1%
8	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Recent Trends in Intelligent Computing and Communication", CRC Press, 2025 Publication	<1%