

Smart Detection of Eggplant Leaf Diseases Using ML and Deep learning

By
Shanzida Sazzad
ID- 213-15-4546

FINAL YEAR DESIGN PROJECT REPORT

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

Supervised by

Ms. Hasnur Jahan

Lecturer

Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

Ms. Nasima Islam Bithi

Lecturer

Department of Computer Science and Engineering
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

September 17, 2025

APPROVAL

This Project titled "**Smart Detection of Eggplant Leaf Diseases Using ML and Deep-learning**," submitted by **Shanzida Sazzad** 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 **17 September 2025**.

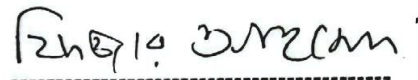
BOARD OF EXAMINERS



Dr. Arif Mahmud

Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

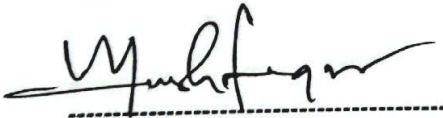
Chairman



Dr. Fizar Ahmed

Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

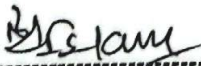
Internal Examiner



Md. Mushfiqur Rahman

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 **Ms. Hasnur Jahan**, Lecturer, 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:

Hasnur Jahan
9.25

Ms. Hasnur Jahan

Lecturer

Department of Computer Science and Engineering

Daffodil International University

Co-Supervised by:

Ms. Nasima Islam Bithi

Lecturer

Department of Computer Science and Engineering

Daffodil International University

Submitted by:

Shanzida

019/25

Shanzida Sazzad

Student ID: 213-15-4546

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. I'm deeply grateful to everyone who has assisted me in one way or another.

First, I express my 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.

I am grateful and wish our profound indebtedness to **Ms. Hasnur Jahan, Lecturer**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Computer Vision and Deep Learning** to carry out this project. Her 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.

I would like to express my 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.

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

Finally, I must acknowledge with due respect the constant support and patience of my parents.

ABSTRACT

Eggplant is a popular crop in developing countries and is severely affected by leaf diseases that influence the harvest and the quality. Proper and timely diagnosis is a key to the sustainability of crop management. This paper explores how contemporary deep learning models can be used to automatically label four typical eggplant leaf states including Leaf Spot, Mosaic Virus, Insect Pest and Healthy. A comprehensive dataset was created by merging 1,008 field-collected images with publicly available images. The manually curated dataset contained 2,032 images, where duplicates, poor images, and incorrect labels were removed and stratified by splitting (70% training, 20% validation, 10% testing) was presented. We compared three architectures: Vision Transformer (ViT), YOLOv8n, and YOLOv11n using the same training protocols in the local and the cloud. They are Precision, Recall, F1-score and mean Average Precision (mAP). YOLOv8n was the most balanced and robust model to deploy in real time, with an overall precision of 0.79, recall of 0.80, F1-score of 0.81, and mAP50 of 0.87. Compared to these, YOLOv11n had high recall rates (0.87) and mAP50 (0.86) with marginally low precision (0.72) whereas ViT had high class-wise F1-scores but poor object-level localization and low run-time performance. YOLOv8n was the most optimal choice between detection and inference speed, suitable in real field practice. This article shows that lightweight detection frameworks, such as YOLOv8n, can play a crucial role in disease surveillance and diagnosis, which can facilitate early intervention and support sustainable agriculture by using AI-based mobile and edge deployment.

Table of Contents

Approval	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
List of Figures	vii
List of Tables	viii
1 Introduction	1-6
1.1 Introduction.....	1
1.2 Motivation	3
1.3 Objectives	4
1.4 Methodology	4
1.5 Project Outcome.....	5
1.6 Organization of the Report	5
2 Background	7-21
2.1 Introduction.....	7
2.2 Literature Review	8
2.3 Gap Analysis	20
2.4 Summary	21
3 Research Methodology	22-29
3.1 Methodology	22
3.1.1 Overview	22
3.1.2 Proposed Methodology	22
3.2 Detailed Methodology and Design.....	24
3.3 Project Plan	27
3.4 Task Allocation.....	28
3.5 Summary	29
4 Implementation and Results	30-40
4.1 Environment Setup	30
4.1.1 Hardware	30
4.1.2 Core software & frameworks.....	30
4.2 Comparative Analysis	31
4.2.1 ViT	31

4.2.2	YOLOv8n	32
4.2.3	YOLOv11n	34
4.3	Results and Discussion.....	37
4.3.1	Implementation of Eggplant Disease Detection System	38
4.4	Summary	39
5	Engineering Standards and Desing Challenges	41-50
5.1	Compliance with the Standards.....	41
5.1.1	Software Standards	41
5.1.2	Hardware Standards	41
5.2	Impact on Society, Environment and Sustainability.....	42
5.2.1	Impact on Life.....	42
5.2.2	Impact on Society & Environment.....	42
5.2.3	Ethical Aspects	43
5.2.4	Sustainability Plan.....	43
5.3	Project Management and Financial Analysis.....	44
5.4	Complex Engineering Problem.....	45
5.4.1	Complex Problem Solving	45
5.4.2	Engineering Activities	48
5.5	Summary	49
6	Research Methodology	51-52
6.1	Summary	51
6.2	Limitation	52
6.3	Future Work	52
	References	53 - 56

List of Figures

3.1	Proposed Methodology diagram	23
3.2	Vision Transformer (ViT) Model Architecture	25
3.3	YOLOv8n Model Architecture.....	26
3.4	YOLOv11n Model Architecture.....	27
4.1	F1-Confidence Curve and Precision-Confidence Curve(YOLOv8n).....	32
4.2	Precision-Recall Curve and Recall-Confidence Curve(YOLOv8n).....	33
4.3	Confusion Matrix (YOLOv8n)	33
4.4	Confusion Matrix Normalized (YOLOv8n).....	34
4.5	F1-Confidence Curve and Precision-Confidence Curve(YOLOv11n).....	35
4.6	Precision-Recall Curve and Recall-Confidence Curve(YOLOv11n).....	35
4.7	Confusion Matrix (YOLOv11n)	36
4.8	Confusion Matrix Normalized (YOLOv11n).....	36
4.9	Image Upload Interface	38
4.10	Disease Detection Results	39
5.1	Project Management Gantt Chart	44

List of Tables

2.1	Summary of Literature Reviewed.....	16
2.2	Gap Analysis Table.	20
3.1	Task Allocation Timeline.....	28
4.1	Test Results of ViT Model	31
4.2	The testing result of YOLOv8n Model	32
4.3	The testing result of YOLOv11n Model	34
5.1	Financial Analysis table.....	45
5.2	Mapping with Complex Engineering Problem.....	46
5.3	Mapping with knowledge Profile.....	47
5.4	Mapping with complex engineering activities.....	48

Chapter 1

Introduction

This chapter presents the research study by giving the background, motivation and objective of the work. It describes the issue under consideration, the approach taken and the anticipated results. Lastly, it introduces the structure of the whole report and provides the reader with an outline of the further chapters.

1.1 Introduction

Food security in most parts of the world is based on agricultural productivity. Eggplant (*Solanum melongena*) is one of the crops grown and it is one of the crops of great nutritional and commercial importance. But the diseases and pest attacks always pose a threat to its production, both in its quality, and yield. The fact that leaves are the most visible and the most essential organs compared to others in the body when it comes to photosynthesis makes them the most key target of early detection system [1]. The manual examination of disease, traditionally depended on by farmers or other agricultural experts, is a time-consuming procedure that is prone to errors particularly when applied at large-scale cultivation fields [2]. This process is complicated further by the inaccessibility of experts within farming communities in remote areas, which typically results in delayed interventions and massive losses in yields [3]. New opportunities to overcome these limitations in agriculture have emerged with the development of computer vision and artificial intelligence (AI). As the number of affordable digital imaging devices continues to increase, and machine learning expands, automated plant disease detection is an exciting field of research [4]. Specifically, the access to curated image datasets has allowed researchers to use advanced models to classify diseases and has opened the door to systems that can be utilized in actual farming conditions [5]. Nevertheless, there are still challenges such as variability in data sets, environmental noise, and the necessity to have a model that is both accurate and computationally efficient [6]. The traditional methods of plant disease recognition have mostly been influenced by conventional convolutional neural networks (CNNs) which is well suited to extract hierarchical visual features of leaf images [7]. The models have been proven successful in processing massive data, however, most of them work on the basis of local receptive fields, which restricts them

to provide global contextual data [8]. Transformer based architecture, including the Vision Transformer (ViT), has been added to computer vision tasks in recent years. As these models use self-attention mechanisms to work on images, there is a potential to extract relationships over the whole leaf surface, such as in complex or noisy situations [9]. Their capability to learn worldly dependencies free of hand-crafted features makes them a good substitute to conventional CNN-based approaches [10]. Meanwhile, object-detecting systems like You Only Look Once (YOLO) have become potent real-time disease detectors. YOLO systems are a hybrid of detection and classification that share a single pipeline, allowing quick predictions to be made on the field [11]. The design of lightweight forms has also increased their utility on resource-limited systems, including mobile phones or edge hardware [12]. The new developments underscore the need to compare various architectural paradigms in order to capture their merits and shortcomings in the farming context [13]. Although the AI-based plant disease detection has made significant advances, there is still a gap in the systematic analysis of transformer-based and YOLO-based architecture in the context of classifying eggplant leaf disease. Numerous existing works are using CNN-based transfer learning or YOLO-derived variants and other crops, and fewer explore the relative effectiveness of ViT and YOLO on eggplant in particular [14]. Moreover, there are other environmental factors like light imbalance, background noise, and duplication of symptoms of various diseases that add some complexities that the models need to consider to be applicable in real life situations [15]. To fill these gaps, it is necessary to develop careful experiments that can not only benchmark models in terms of accuracy but also in terms of robustness, interpretability, and efficiency [16]. Thus, a direct comparative study of Vision Transformers and YOLO architectures to detect the eggplant leaf disease is not made in this project. The majority of current literature is dedicated to CNN-based classifiers or lightweight YOLO variants, but little is done to investigate transformer-based models in the agricultural sphere. Moreover, the issues related to imbalance in classes, overlapping symptoms (i.e., leaf spot and mosaic virus), and working in natural conditions have not been solved yet. This paper fills such gaps by carrying out a systematic comparison of ViT, YOLOv8n and YOLOv11n to a curated eggplant dataset to determine which model is most effective in practice to detect disease and deploy to the field.

1.2 Motivation

The intensive growth of world agriculture reveals the high priority of smart and automated systems that will reduce the losses that result due to plant diseases. Eggplant, as most crops are susceptible to numerous types of leaf diseases that can seriously diminish the harvest and quality [1]. The conventional diagnostic methods are still reliant on expert opinion and manual scrutiny that are very subjective, labor intensive and impossible to perform on a large scale [2]. This constraint offers solid computational incentive to use artificial intelligence and computer vision solutions to support farmers with scalable and trusted disease surveillance. As digital datasets and sophisticated machine learning have become a reality, the possibilities of creating automated recognition frameworks have increased [4]. These systems are capable of analyzing large amounts of leaf-image and identifying disease patterns which are not readily discernible by the human eye [5]. But there are obstacles to overcome like changing environmental conditions, sparse annotated data, and disease presentation disparities that demand more than just an accurate model; they demand a robust and generalizable model [6]. With recent progress in real-time detection algorithms, such models are now able to be implemented in mobile and embedded designs [12]. Such portability means that disease recognition systems can be deployed at the field, where farmers can immediately receive feedback without a centralized laboratory analysis. Moreover, lightweight detection models can be successfully deployed in resource-limited rural settings, which increases the feasibility of such solutions. In addition to technical innovation, this work is driven by the need to address the gap between the state of the art research in deep learning and the application of deep learning technology in agriculture. Although most studies consider CNN-based models, few have dared to investigate the comparative benefits of transformer-based models with detection-based models in similar agro settings [14]. It is necessary to make this comparison in order to find out which architecture is better adapted to the complexities of eggplant disease detection with trade-offs in precision, recall, and computational cost. Finally, the resolution of this issue is of great social and economic value. It allows farmers to detect diseases early and in doing so, reduce the use of pesticides, decrease the cost of production, and decrease yield loss. At a larger scale it is involved in food security, sustainable farming and enhancement of livelihoods of the farming communities. The computational innovations and agricultural necessity give a powerful impulse to read and investigate Vision Transformers and YOLO frameworks to identify the eggplant leaf disease in the image.

1.3 Objectives

The principal aims of this study are listed below:

- I. To create a curated dataset of eggplant leaf images with four categories: Healthy, Insect Pest, Leaf Spot and Mosaic Virus.
- II. Develop and train deep learning models (Vision Transformer and YOLO alteration) in the same conditions of the experiment and demonstrate their fairness.
- III. A deployable prototype containing an application programming interface (API) and a minimal-scale web or mobile user interface (UI) that provides real-time predictions and actionable guidelines (treatment/prevention) specifically designed to be used on farms should be a deliverable

1.4 Methodology

This research has a systematic workflow that will guide the methodology to provide accurate assessment and fair comparison of the various architectures of deep learning in detecting the disease on eggplant leaf. The first step is preparing the dataset, during which the pictures of the leaves of eggplant are taken both in the field and from the internet. Data cleaning and preprocessing such as background removal, resizing, and normalization are then performed to increase consistency and emphasize disease features. After the preprocessing, the dataset is stratified into training and testing datasets to maintain the distribution of classes and to allow robust evaluation. Three models are chosen to be experimented: Vision Transformer (ViT, vit_base_patch16_224) is selected as a transformer-based image classifier, YOLOv8n is selected as lightweight real-time image detector, and YOLOv11n is the most recent model of the YOLO series with the improved feature extraction ability. All models are trained in similar conditions on the same dataset to provide a fair benchmark. The widely accepted measures of performance evaluation are the accuracy, precision, recall, F1-score and mean Average Precision (mAP). Such measurements enable to compare not just the classification capabilities of each model but also their efficiency and their ability to operate in real-time. Lastly, a comparative study is conducted to bring out the trade-offs between ViT and YOLO models in accuracy, recall and computation efficiency. The results are informative about the choice of the most efficient architecture to detect the eggplant disease, and can be extended to other crops and integrated into agricultural decision-support systems.

1.5 Project Outcome

The results of this study should have a scientific and practical use in the sphere of intelligent agriculture. One of the main outcomes of this work will be the overall performance benchmark of Vision Transformer (ViT), YOLOv8n, and YOLOv11n with explanations of their relative strengths and weaknesses in application to the eggplant leaf disease detection. Through the analysis of these architectures, it is hoped that the study will determine the most appropriate architecture that best balances the classification accuracy, recall, and computational efficiency, rendering it practical in real-time agricultural usage. The second by-product of this work is the creation of a scalable dataset preparation, model training and evaluation methodology that can be modified to other crops and plant disease recognition cases. Another focus of the research is the practical potential of implementing lightweight and efficient models with mobile or edge devices so that farmers can detect diseases in the field without having to specially support it in the laboratory. Other than technical contributions, the work is likely to enhance sustainable agriculture in terms of early disease prevention, minimizing unnecessary use of pesticides, minimizing production expenditures and losses of yields. All these are meant to enhance food security, as well as to enhance environmentally friendly and economically viable agricultural production.

1.6 Organization of the Report

The report is presented in six chapters that cover a particular part of the study in a logical order. Chapter 1 provides an introduction to the research, including the background, motivation, objectives, methodology and outcomes to be expected, including how the report is formatted. Chapter 2 contains background and related research information about the detection of plant diseases and offers an overview of the existing methods, datasets and techniques. An analysis of gaps is also provided within this chapter that highlights the weaknesses of the previous research and locates the contribution that the proposed research would provide. In Chapter 3, the research methodology is outlined and provides a summary of the approach, the recommended workflow, specifics of the data preprocessing, the model selection, and the evaluation strategy, and a project plan and schedule of tasks. Chapter 4 is devoted to implementation and its outcomes, detailing the experimental setting, reporting the results of model training and evaluation and making a comparative analysis and a discussion of the obtained results. In chapter 5, the authors cover engineering standards and design issues, such as adherence to software and

hardware standards, societal and environmental effects, project management, and mapping with complex engineering issues and undertakings. Lastly, Chapter 6 completes the report with a summary of the key findings, limitations of the research and recommendations on future research. All these chapters contribute to a logical progression of the problem definition through solution development, evaluation, and conclusion.

Chapter 2

Background

This chapter presents the background necessary to understand the research problem and locates the study within the broader discipline of artificial intelligence in agriculture. It starts with an introduction in which the importance of plant disease detection is discussed, and how computer vision is used to solve this problem. Then it provides literature review of related works including some key methods and data sets that were examined in past research. A gap analysis is then presented identifying shortcomings from existing approaches and explaining the need for the proposed system. Finally, a summary that integrates the lessons learned from the reviewed studies is presented as the chapter's conclusion.

2.1 Introduction

The importance of early disease detection in agricultural systems for crop productivity and food security has driven research in agricultural engineering and computer vision. Like many vegetable crops, eggplant is susceptible to a myriad of leaf diseases, some of which are among the most common yield and quality disorders. Conventional diagnosis relies heavily on manual inspection, which is time-consuming, subjective and unsuitable for production systems with large-scale production. These restrictions have promoted the use of automated methods based on image analysis coupled with artificial intelligence to assist farmers in decision making in time [1]. With the growing availability of digital imaging devices and the development of machine learning algorithms, the advances in plant disease recognition systems have been accelerated [2]. Using leaf image datasets, it has been possible to train models that can discriminate between healthy and diseased samples, even under more challenging conditions [3]. Especially deep learning has been demonstrated to be a revolutionary technology, which allows the automatic extraction of discriminative features without the use of handcrafted rules [4]. It is, therefore, well suited for agricultural applications where traditional image processing methods are complicated by the variability in light, background and leaf morphology [5]. While deep learning models are primarily being trained to improve classification accuracy, there has been a growing interest in optimizing deep learning models for speed and

deployment efficiency. This is even more important for mobile and edge-based applications where farmers can take advantage of on-site disease detection using lightweight models [6]. At the same time, the advent of transformer-based architectures has opened new possibilities for image analysis, providing mechanisms for reasoning about global dependencies on the whole input rather than isolated patterns. These advances make it possible that a detailed comparison of alternative architectures can offer valuable insights for their applicability for real-world agricultural systems [7].

This chapter introduces the basis for such a comparison through a review of representative related works, an analysis of how different models have been applied to leaf disease detection, and an investigation of the pros and cons of each. In addition to summarizing the evolution of CNN- and YOLO-based methods and the emergence of interest in transformer-based approaches, the gap analysis helps identify the specific contributions required to bring the state of the art in eggplant disease detection forward.

2.2 Literature Review

Howlader et al. [1] developed an extensive eggplant (brinjal) leaf image database of multiple disease types, including healthy plants, for data-driven agricultural research. The dataset, which is mostly captured by smartphone cameras under controlled and semi-controlled environments, standardizes class labels and image quality to mitigate annotation noise and allow reproducible benchmarking. Simple pre-processing (cropping, color normalization) and naming standardization were emphasized to increase the reliability of downstream training. Even though the paper is more data-set focused (it doesn't present a novel model), it created a standardized basis for fair comparisons between deep learning pipelines. A major limitation is that conditions used for capture may be domain biased, as controlled capture conditions do not fully capture complex environments in the field.

Liu and Wang presented [2] "EggplantDet", a lightweight disease detector based on the YOLO family and optimized for the deployment efficiency. Their pipeline incorporates standard augmentations (flips, scale, color jitter) and mosaic techniques to make it more robust. The backbone is compact and attention is used in combination with a refined IoU-based loss to stabilize training. The optimal setup obtained competitive mAP while maintaining real-time throughput on edge-class GPUs, and is

therefore suitable for deployment in the field. The only limitation is external validity: performance is still sensitive to extreme lighting changes and unknown backgrounds which indicates the need for future work on domain generalization and field augmentation.

Paymode and Malode [3] investigated transfer learning for crop leaf disease detection based on CNN backbones like VGG on public datasets. The study used standard pre-processing (resized to canonical input, normalization) and medium augmentation. The finely tuned VGG variants gave excellent single-image classification accuracy, demonstrating the viability of pretrained CNNs with limited labeled data. The best model provided high accuracy using curated datasets but the authors mention issues with cross-domain generalization (from lab-like images to field with occlusions and changing illumination conditions).

Abisha and Jayasree [4] proposed an intelligent system by integrating the image processing (segmentation to isolate lesion regions) with ANN classifier for early detection of eggplant disease. Their dataset consisted of images of leaves taken under mixed conditions and then preprocessed (denoised, color-space conversions) to improve the contrast of the lesion. The ANN pipeline resulted in better recognition than classical baselines, especially for visually more distinct lesions. The major drawback is scale: the performance was reported on a relatively small data-set and generalization to different agro-ecological systems requires more varied training corpora and stronger regularization.

Kursun and Koklu [5] studied a hybrid in which AlexNet is used to extract deep features followed by classification with a Random Forest. With multiple eggplant diseases and healthy leaves, the dataset was then resized and normalized according to the standard pipeline. The highest performing configuration highlighted the importance of deep features combined with a powerful shallow classifier in low data regimes. This two-stage framework combined expressive representation with regularized decision boundaries. However, deep backbone features were relied upon and field diversity was limited, which impaired robustness, suggesting a need for ensembles and wider augmentation.

Islam et al. [6] concentrated on the local identification of spinach leaves building a

dataset of several thousand images across common varieties of spinach. The images were pre-processed (clean, resized and normalized); class balance was corrected using augmentation. VGG16 provided the highest classification accuracy of evaluated CNNs, indicating transfer-learned convolutional features are very effective for leafy-vegetable phenotype recognition. The authors warn that performance is likely to be inflated by relatively controlled backgrounds and fixed perspectives, and they call for validation on more challenging field-captured distributions.

Haque and Sohel [7] presented a fusion pipeline using the CNN derived features with SVM or softmax classifiers for detection of eggplant leaf blight and fruit rot. It was demonstrated that the system provided improved discrimination over single-model baselines after preprocessing (contrast enhancement, resizing). The best fusion variant produced better class separation, presumably because of complementary decision functions across feature spaces. However, the limited geographical diversity and class imbalance were identified as limitations of the dataset and motivated targeted collection of data and cost-sensitive learning for dealing with underrepresented symptoms.

Orchi et al. [8] performed a comparative study of traditional ML (SVM, kNN, RF) and transfer learning (VGG, ResNet, Inception) on plant-disease datasets. Inputs were scaled, normalized, and moderately augmented. The deep transfer-learning models always performed better than the classical ML, especially on larger curated image sets. A significant result is that the pretrained CNNs outperformed the handcrafted features in generalizing well within a dataset, but the handcrafted features suffered from visual variability. The limitation was over-reliance on curated benchmarks: generalization to the real world is the central problem the authors identify.

Tan et al. [9] systematically compared ML and DL methods for classification of tomato leaf diseases based on a standardized data set and pre-processing pipeline. Using deep architectures, and in particular mid-depth residual networks, we have shown close to ceiling accuracy under controlled conditions, outperforming SVM/kNN baselines. The work provides further justification for end-to-end feature learning. However, the work highlights a recurring issue, namely that a high in-domain accuracy does not imply robustness to field noise (shadow, clutter, occlusion), which in turn spurred requests for domain adaptation and cross-domain evaluation

protocols.

Ngugi et al. [10] gave a general survey of ML and DL applied to leaf disease detection of plants. Datasets, preprocessing (including color-space transforms and lesion-segmentation strategies), model families ranging from CNNs to attention-based hybrids were summarized in the review. A key takeaway is that deep networks continue to dominate on curated images, and that there remain challenges of label noise, imbalance, and domain shift. The survey highlights promising directions such as lightweight models for deployment at the edge and multi-label recognition for co-occurring diseases; limitations include that benchmark standards between studies are unevenly distributed.

Ajra et al. [11] tested deep CNNs (e.g., AlexNet, ResNet50) for tomato and potato leaf disease classification using common pre-processing and augmentation. They found that the accuracy of moderately deep CNNs is high under regularised conditions. The research paper highlighted the importance of hyperparameter tuning and data augmentation towards stability. Limitations include possible overfitting to artifact-specific features of the dataset and the need for systematic tests on uncurated field captures to assess real world readiness.

In Xie and He [12], hyperspectral imaging (380-1023 nm) was studied for early detection of eggplant blight, extracting spectrum-texture features (e.g., GLCM) and training kNN and AdaBoost classifiers. Spectral calibration and denoising were performed as preprocessing steps; segmentation was used to determine regions of interest in which to calculate features. Best classical pipeline showed good early discrimination with an added value over RGB. However, specialized sensors and acquisition control restrict the scalability and the approach needs cautious domain adaptation to generalize across devices and field conditions.

In one work, Ferentinos [13] trained deep networks on a large multi-crop training set (~88k images across dozens of species/classes), with standard image normalization and augmentation. Extremely high accuracy for CNN-based single image classification confirmed the feasibility of CNN-based disease phenotyping. The broad scope of the study is an advantage, but it suffers from the same limitation as in the field of image processing: curated imagery tends to be free of clutter and variation

present in real world scenes, and results should thus be interpreted as an upper bound on achievable real-world performance.

Nahiduzzaman et al. [14] presented PDS-CNN for leaf disease classification with explainability using SHAP, which allows the visualization of attribution of features at inference time. The dataset was composed of preprocessed images of mulberry leaves that increases the contrast between the lesions. The model also achieved high binary accuracy and provided interpretable saliency maps for practitioner trust. The main contribution is explainable inference; the limitation is lower complexity of the inference compared to the multi-class, multi-disease counterpart and the need to verify explanations under real world noise.

Bonkra et al. [15] performed a scientometric study of deep learning for apple leaf disease detection including a synthesis of publication trends, datasets and model choices from 2011 to 2022. While not an empirical benchmark, the work does point out the direction from handcrafted features to CNNs and the new emerging transformer-based techniques. The review reflects growing interest in deployment efficiency and dataset realism. Its limitation is inherent to meta-analyses: model testing is not possible, and so conclusions on performance are based on reported results in disparate studies.

Wei et al. [16] published PlantSeg, a large-scale, in-the-wild dataset that contains pixel-level annotations for plant disease segmentation in different environments. The data set documentation provides information on acquisition variability (lighting, occlusions) and annotation protocols to achieve mask quality. The contribution is to allow segmentation benchmarking for not only classification-only datasets. As a dataset paper, it does not restrict users to one best model; the only limitation is that there are still distributional shifts across the geographies and crops, necessitating careful splits and cross-site validation.

Wang et al. [17] applied a multimodal fusion method, which combines image and auxiliary signals with the use of an attention mechanism to identify the presence of eggplant disease. Preprocessing aligned features and normalized inputs; features at intermediate levels were fused by the network to enhance robustness. The most successful model was highly accurate and better stable at varying conditions. The

complexity is the trade-off: sensor integration, synchronization, and calibration increases the deployment cost, and practical implementations require a meticulous cost-benefit consideration.

Jayanthi, & Shashikumar. [18] introduced an eggplant leaf analysis pipeline which initially divides up diseased regions by adaptive multi-kernel fuzzy C- means (ARMKFCM) and then classifies with an optimized probabilistic neural network (PNN) optimized by a metaheuristic. Denoising and color normalization were part of preprocessing to stabilize clustering. The method enhanced the lesion segregation and downstream categorization. Disadvantages are that computational overhead in iterative clustering is high, and sensitivity to hyperparameters encourages approximate efficient methods to be practical in real-time applications.

Abisha et al. [19] applied discrete shearlet transforms to extract multiscale features, using a deep CNN and an RBF neural network as a comparison to multiple eggplant diseases. The dataset (about 1.5k images) was resized and normalized regarding illumination. The CNN variant with deep architecture gave the highest accuracy and RBF baseline followed. The important point is that multiscale representations increase the fine-scale lesion textures; the constraints include the medium size of datasets and possibly overfitting without cross-domain testing.

Xie et al. [20] introduced YOLOv5s-BiPCNeXt, a small detector that has the MobileNeXt-like backbone and a bi-directional neck designed to fit the symptoms of the eggplant diseases. They had class-specific annotations in the training set; stability was improved by augmentation and label smoothing. The most successful model was associated with high AP between disease classes without compromising about real-time FPS on embedded GPUs. Its drawback is the standard accuracy-speed trade-off: performance can be limited by gains in efficiency on small lesions or in very high-light without further refinement.

Zhang et al. [21] aimed to detect *Verticillium* wilt in eggplant seedlings at an early stage through inexpensive multispectral channels fused to a deep CNN (last variant with attention). The data were recorded at an early stage of symptom onset; spectral normalization and ROI extraction were performed. The optimum setup obtained good accuracy in screening early. This weakness is due to scope: the model is trained to a

particular pathogen and growth stage, and would need retraining or domain adaptation to cover a broader range of diseases and conditions of mature leaves.

Rangarajan and Purushothaman [22] assembled one of the first eggplant disease datasets (five classes) and tested both end-to-end VGG16 and a hybrid VGG16 feature extractor on a multi-class SVM. Laboratory and field images were resized and analyzed in various color spaces. Accuracy using VGG16+MSVM pipeline was very high on the curated set. The authors observe that the high in-domain performance should be confirmed with field variability, which suggests augmentation and cross-site testing to avoid over-fitting.

Kaniyassery et al. [23] examined the relationships of meteorological variables with the incidence of leaf spot and fruit rot in an eggplant variety with GI tags and produced Leaf Guard, an Android application that combines an AI model to detect disease ROI. The dataset was a mixture of field images and other related weather records; pre-processing was used to keep time and image data parallel. The integrated model (YOLO-based) showed good test accuracy in region localization. One of them is regional specificity- not only climate patterns, but cultivar traits, may restrict transfer, so site-conscious calibration is essential.

Cemek et al.[24] applied the ML (ANN, RF, SVM, DNN) to predict evapotranspiration (ET) of eggplant using agricultural time series. Even though the study is not a disease dataset, it represents how agronomic decision support handles data preprocessing (missing-value handling, normalization) and supervised learning. The optimal ANN showed high R² to estimate ET, which demonstrates the benefit of using ML in irrigation planning. It is relevant here in a methodological sense: efficient pipelines to preprocessing and validation can be translated into plant-health analytics; its limitations are site-specificity common to agrometeorological models.

Demilie [25] compared methods of plant disease detection and classification in both ML and DL and tabulated patterns in performance and evaluation. The review highlights that CNNs can learn spatial hierarchies directly on images and highlights some long-standing challenges, including imbalanced datasets, limited generalization between crops/sites, and non-standard evaluation. It promotes the use of multi-label recognition and more realistic and real-world data. The weaknesses comprise the use

of reported (and not always comparable) measures, which contributes to the importance of community benchmarks with regular splits and field validation.

Sapkota et al. [26] compared one-stage YOLOv8 to two-stage mask r-cnn in respect to instance segmentation of complex orchard scenes. Two instance-level annotated datasets (dormant branches; green fruitlets) were preprocessed by tiling and normalization. YOLOv8 provided extra accuracy/recall and reduced inference latency compared to Mask R-CNN, and was more suitable to real-time agricultural robotics. Its weakness is ecological specificity (apple orchards), but the methodological lesson, the speed of one-stage models with competitive accuracy, can be applied successfully to crop-disease lesion segmentation tasks.

Hassan et al. [27] introduced a wheat yellow rust field-ready pipeline based on YOLOv8 leaf localization, an unsupervised rotation step to maximize the area of leaves in bounding boxes, segmentation models (UNet, DeepLab variants, SAM), and classification by CNNs and vision transformers. UNet had the highest segmentation IoU in natural backgrounds and a Swin Transformer had the highest classification accuracy. The tradeoff is a powerful, end-to-end workflow; the weakness is agro-ecological crop-specific tuning and the requirement to test against a more agro-ecologically representative scene.

Haider et al. [28] proposed the EDFS-Net and MDFS-Net based on the plant-disease and fruit segmentation, focusing on multi-scale multi-receptive-field fusion with the effective grouped convolutions. Databases consisted of coffee leaf, cassava, and orchard fruit; preprocessing was normalized resolution and illumination. The last model recorded high Dice scores and required an order of magnitude fewer parameters than some of the baselines, which is an efficiency improvement. Limitations will include loose generalization to species not evaluated, the cost of careful architectural tuning to new domains.

ADCLR: Attention-based dilated CNN feature extraction with logistic regression classification on tomato leaf diseases was proposed by Islam et al. [29]. A curated PlantVillage split was used as the dataset; bilateral filtering and Otsu segmentation were performed, and a CGAN was used to balance. The method produced extremely high benchmark accuracy, showing that lightweight classifiers may be used with

discriminative features. Probably the constraint is domain shift; again performance should be re-tested on field imagery including clutter and arbitrary light.

Islam, Raisa, and Khan [30] discussed leafy vegetables in Bangladesh developing two datasets: one to be used as a species classification dataset and other to be used as a disease instance segmentation dataset. Standardized size and noise were preprocessed; detection/segmentation experiments were conducted with VGG16/19, ResNet50 and YOLOv5/7/8. YOLOv8 variants tend to provide an optimal segmentation-speed trade-off across diseased leaves. Most important is local data and dual-task assessment; coverage breadth is one weakness more sites and seasons would bolster generalization to deployment at national scale.

In below, Table 2.1 shows the summary of literature review.

Table 2.1: Summary of Literature Reviewed.

Author (s)	Title	Dataset Description	Methodology	Key Findings and Limitations
Howlader et al., 2025 [1]	Eggplant Leaf Disease: A comprehensive image dataset for the identification.	4,089 eggplant leaf images across six classes (smartphone, controlled & semi-controlled)	Dataset creation for benchmarking	High-quality labels; limited field realism
Liu & Wang, 2025 [2]	EggplantDet: A Lightweight Eggplant Leaf Disease Detection Model Based on YOLOv8	Eggplant disease images with labeled classes	YOLOv8-based lightweight model with FasterNet backbone, TAM, WIoU	92.61% mAP, 88 FPS; domain shift issues
Paymode & Malode, 2022 [3]	Transfer Learning for Multi-Crop Leaf Disease Detection Using CNN	Public datasets (PlantVillage grape & tomato)	VGG-based transfer learning	High accuracy; limited cross-domain robustness
Abisha & Jayasree, 2022 [4]	Intelligent System for Early Eggplant Leaf Disease Detection	Eggplant leaves with visible lesions	ANN + image segmentation	Effective early detection; small dataset size

Kursun & Koklu, 2025 [5]	Automatic Detection of Eggplant Leaf Diseases Using AlexNet and Random Forest	Eggplant leaf datasets (multi-class)	AlexNet feature extraction + Random Forest	Hybrid improved accuracy; diversity limitation
Islam et al., 2022 [6]	Automatic Classification of Local Spinach Leaf Species Using Deep Learning	3,785 spinach leaf images across 5 species	CNNs (VGG16, VGG19, Inception, Xception)	VGG16 reached 99.79% accuracy; curated dataset bias
Haque & Sohel, 2022 [7]	Fusion CNN-SVM for Eggplant Leaf Blight and Fruit Rot Detection	2,284 eggplant leaf & fruit images	CNN feature extraction + SVM/Softmax	Fusion improved recognition; imbalance issues
Orchi et al., 2023 [8]	Comparison of Machine Learning and Deep Transfer Learning for Plant Disease Detection	PlantVillage datasets (multi-crop)	SVM, RF, NB vs. VGG16, ResNet50, InceptionV3	DL > ML; curated dataset limits generalization
Tan et al., 2021 [9]	Comparison of ML and DL Methods for Tomato Leaf Disease Detection	Tomato leaf images (PlantVillage)	ResNet34 vs. ML baselines	ResNet achieved 99.7%; lacks field realism
Ngugi et al., 2024 [10]	A Survey of Machine Learning and Deep Learning Methods for Crop Disease Detection	Review of 25+ crop disease datasets	CNNs, ResNet, VGG, hybrids	DL dominance; uneven benchmark standards
Ajra et al., 2020 [11]	Deep CNN Models for Tomato and Potato Leaf Disease Classification	Tomato & potato leaf datasets	AlexNet, ResNet50	Strong accuracy; overfitting to curated datasets
Xie & He, 2016 [12]	Hyperspectral Imaging for Early Detection of Eggplant Blight	Hyperspectral images (380–1023 nm)	GLCM features + kNN, AdaBoost	Early detection feasible; sensor cost limits scaling

Ferentinos, 2018 [13]	Deep Learning Models for Plant Disease Detection and Diagnosis	87,848 leaf images (25 plants, 58 classes)	CNN classifiers	99.53% accuracy; dataset bias acknowledged
Nahiduzzaman et al., 2023 [14]	PDS-CNN with SHAP for Mulberry Leaf Disease Detection	Mulberry leaf images	Custom CNN + explainability (SHAP)	96.06% accuracy; binary focus, limited complexity
Bonkra et al., 2023 [15]	Scientometric Analysis of AI for Apple Leaf Disease Detection	214 papers reviewed (2011–2022)	Bibliometric mapping	Identified DL growth trends; no experiments
Wei et al., 2024 [16]	PlantSeg: A Large-Scale Dataset for Plant Disease Segmentation	11,400 annotated segmentation masks, 8,000 healthy	Dataset release for segmentation benchmarking	Enables pixel-level tasks; geographic bias persists
Wang et al., 2025 [17]	Multimodal Fusion Model for Eggplant Disease Detection	Eggplant images + sensor data	Attention-based multimodal fusion	High accuracy & robustness; sensor costs limit use
Jayanthi, & Shashikumar, 2023 [18]	Adaptive Multi-Kernel FCM and Optimized PNN for Eggplant Leaf Diseases	Eggplant disease leaf images	ARMKFCM segmentation + PNN (metaheuristic optimized)	Improved lesion isolation; computationally heavy
Abisha et al., 2023 [19]	Eggplant Leaf Disease Classification Using Discrete Shearlet Transform and DCNN	1,500 eggplant leaf images	Shearlet + CNN vs. RBFNN	CNN outperformed RBF; dataset size small
Xie et al., 2024 [20]	YOLOv5s-BiPCNeXt for Eggplant Leaf Disease Detection	Eggplant leaves annotated per disease	YOLOv5s with BiPCNeXt neck	Strong AP (94–99%); weaker on tiny lesions
Zhang et al., 2024 [21]	Early Detection of Verticillium Wilt in Eggplant Using Multispectral Channels	Seedling images with early symptoms	VGG16 + triplet attention	86.7% precision; pathogen-specific scope
Rangarajan & Purushothaman, 2020 [22]	Disease Classification in Eggplant Using Pre-Trained	5-class dataset (field + lab)	VGG16 features + MSVM	99.4% accuracy; requires validation

	VGG16 and MSVM			in real fields
Kaniyassery et al., 2024 [23]	Association of Meteorological Variables with Eggplant Diseases and AI-Based Detection	Mattu Gulla eggplant leaves + weather data	YOLOv8-based ROI detection + regression	98.2% accuracy; region-specific performance
Cemek et al., 2023 [24]	Machine Learning for Estimation of Eggplant Crop Evapotranspiration	Agrometeorological data	ANN, RF, SVM, DNN	ANN best ($R^2=0.984$); site-specific constraints
Demilie, 2024 [25]	Plant Disease Detection and Classification Techniques: A Comparative Study	Multi-crop survey	CNNs, ML, DL comparisons	CNNs effective; dataset bias & imbalance noted
Sapkota et al., 2024 [26]	Comparing YOLOv8 and Mask R-CNN for Instance Segmentation in Orchards	Apple orchard datasets (branches, fruitlets)	YOLOv8 vs. Mask R-CNN	YOLOv8 superior speed/accuracy; orchard-specific
Hassan et al., 2024 [27]	Wheat Leaf Localization and Segmentation for Yellow Rust Detection	Wheat rust dataset, natural backgrounds	YOLOv8 + UNet + Swin Transformer	IoU=0.9563, Acc=95.8%; crop-specific
Haider et al., 2024 [28]	Multi-Scale Feature Fusion for Plant Disease and Fruit Segmentation	Coffee, cassava, apple fruit	EFFS-Net, MDFS-Net	High Dice scores; model complexity in tuning
Islam et al., 2022 [29]	Attention-Based Dilated CNN with Logistic Regression for Tomato Leaf Diseases	Tomato leaf images (PlantVillage)	Dilated CNN features + LR	96–100% accuracy; domain shift to field images
Islam et al., 2023 [30]	Enhanced Leafy Vegetable Analysis Using CNN and YOLO Models	Leafy vegetables: 3,306 classify, 4,493 disease seg	VGG16/19, ResNet50, YOLOv5/7/8	YOLOv8 best segmentation; limited seasonal data

2.3 Gap Analysis

In the related works review it is shown that deep learning techniques, especially convolutional neural networks (CNNs) and YOLO-based detectors are currently the leading plant disease detection research methods. While these methods have shown good performance on several benchmark datasets, such transformer-based architectures as the Vision Transformer (ViT) are still relatively unexplored in the agricultural domain. In the case of eggplant, a highly disease-prone crop, most of the published research has been focused either on CNN-based transfer learning or on specialized imaging modalities such as hyperspectral sensors. Though these methods give valuable information, they are usually constrained by limitation of dataset, computational cost or high cost of equipment for deployment purposes, which renders them less scalable. A second shortcoming of literature is the limited scope of architectural reviews. Many studies only use one type of model, so it's hard to compare the performance of different kinds of deep learning paradigms under the same setting. Further, performance evaluations are often performed on manually curated datasets acquired in a controlled laboratory setting. Such conditions minimize sources of variability in illumination, background clutter, and symptom overlap which tends to exaggerate reported accuracy but compromise generalizability. Thus, generalization to field conditions is still a major challenge. In the proposed work, we directly bridge these gaps by empirically comparing Vision Transformer (ViT), YOLOv8n, and YOLOv11n against each other on the same curated eggplant dataset. In addition to benchmarking classification accuracy, this paper also assesses real-time applicability and robustness across four different disease classes. By doing so, it should give us valuable insight into what architecture is the best to strike a balance between accuracy, efficiency, and practicality for disease detection systems that can be deployed in the field.

In Table 2.2 shows the features of our proposed system with compare other studies.

Table 2.2: Gap Analysis Table.

Features	Liu & Wang [2]	Xie & He [12]	Xie et al. [20]	Rangarajan & Purushotham [22]	Kaniyassery et al. [23]	Proposed System
Eggplant-specific focus	Yes	Yes	Yes	Yes	Yes	Yes

RGB image-based detection	Yes	No	Yes	Yes	Yes	Yes
Use of Transformer models	No	No	No	No	No	Yes (ViT)
Multi-class classification	Yes	No	Yes	Yes (5 classes)	Yes	Yes (4 classes)
Real-time applicability	Yes (88 FPS)	No	Yes (~26 FPS)	No	No	Yes (YOLOv8/11)
Field robustness	No	No	No	Limited	Region-specific	Enhanced through preprocessing

2.4 Summary

This chapter served the purpose of setting the right context for understanding the topic of eggplant leaf disease detection with deep learning. A comprehensive literature survey of thirty related works showcasing the advancements of CNN- and YOLO-based approaches as well as the emergence of specialized imaging and hybrid approaches. While these existing works achieve promising performance on curated datasets, most of them suffer from low datasets diversity, cross-architecture comparison and low applicability in the real field. In the gap analysis, we found that the usage of transformer-based models like ViT is seldom studied in this context and that common comparisons to YOLO are still lacking. To overcome these shortcomings, the proposed research compares ViT, YOLOv8n and YOLOv11n on the same dataset in terms of multi-class classification, real-time suitability and robustness. This comparative framework serves to provide a basis for understanding the most suitable architecture for performing scalable agricultural applications.

Chapter 3

Research Methodology

This chapter explains the methodological framework used for the study. The paper starts with an introduction to the research workflow followed by the proposed methodology presented in the form of a structured pipeline diagram. The chapter then outlines the processes for collecting, pre-processing and training and evaluating datasets using ViT and the two leading YOLO architectures, YOLOv8n and YOLOv11n. Finally, it describes the project design and assignment of tasks to show how the research was systematically planned and carried out.

3.1 Methodology

3.1.1 Overview

The aim of this research is to perform comparative study of modern deep learning architectures for eggplant leaf disease detection. The key point is an evaluation of the performance of transformer-based models defined by the Vision Transformer (ViT) compared with YOLO-based architectures (YOLOv8n and YOLOv11n) which are optimized for real-time detection. The study starts with the gathering of an eggplant leaf dataset, which includes the healthy and diseased samples of four target categories: Healthy, Insect Pest, Leaf Spot and Mosaic Virus. These images are then curated and prepared to create a reliable data set to train and test. After completing the dataset, we train and test the three selected models under the same experimental conditions. Their performances are compared with standard evaluation metrics like accuracy, precision, recall, F1-score, mean Average Precision (mAP) etc. From these results the trade-offs between accuracy and efficiency among the models are intended to be discerned. Lastly, the developed methodology ensures that the research not only compares those architectures on the same dataset, but also offers practical implications regarding the suitability of those architectures for real-time agricultural applications.

3.1.2 Proposed Methodology

The proposed methodology is organized into a structured pipeline for consistent data handling, training and evaluation for the detection of eggplant leaf diseases. The workflow

starts with the collection of the samples by directly taking the image from the field observations by digital devices. To further improve dataset diversity, new images from public repositories like Mendeley are used to combine images from the real-world field with public data, the dataset is balanced across four classes and divided into a training set and a test set, training three chosen models on the training set and testing their performance on the test set. The final outputs are categorised in four target classes. As Figure 3.1 Proposed Methodology shows, this pipeline not only allows fair benchmarking across architectures but it also provides a structured framework for comparative analysis.

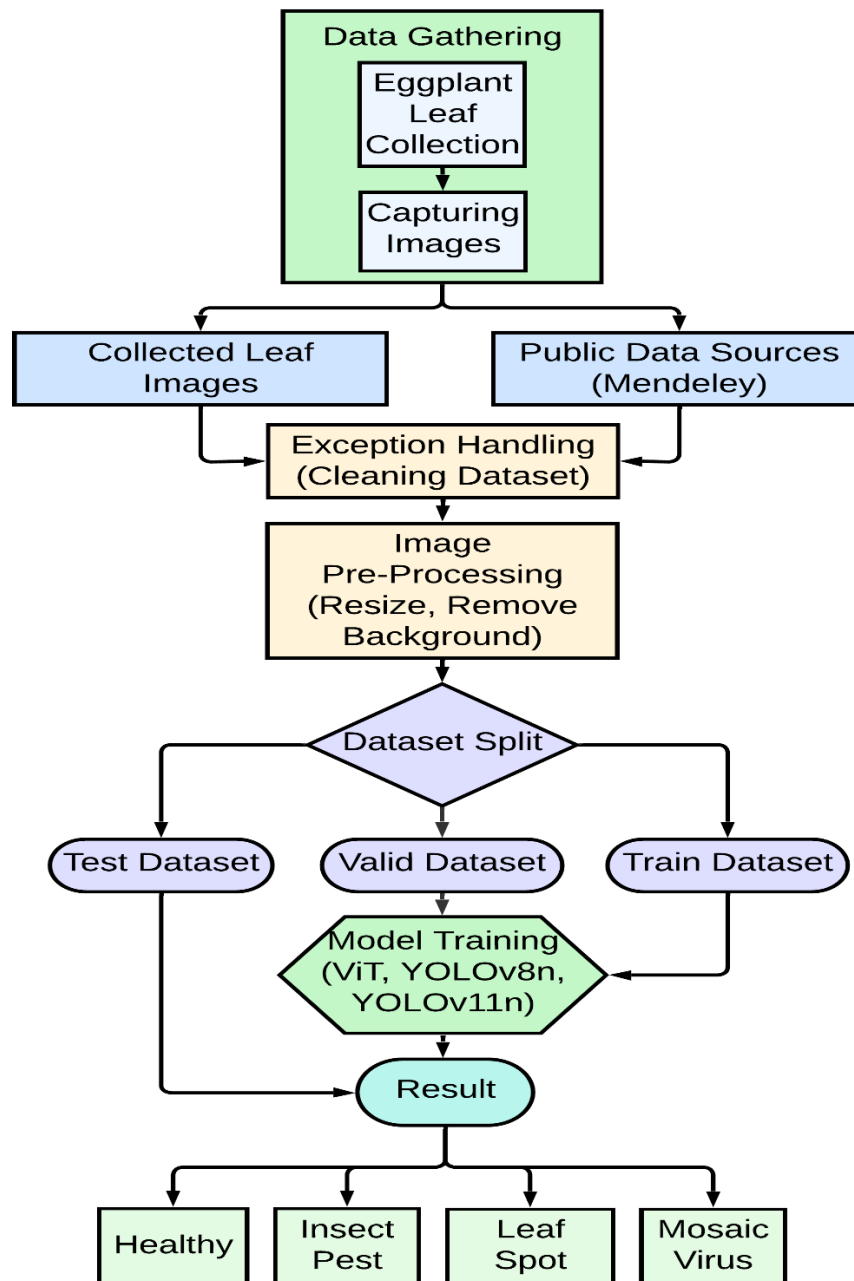


Figure 3.1: Proposed Methodology diagram

3.2 Detailed Methodology and Design

Data Pre-processing

The dataset used in this study went through a well-defined pipeline of preprocessing to make sure that all the models were built on the same data. Once the images from the field sources and public repositories were obtained, manual cleaning was performed to eliminate redundant, blurred and irrelevant data. The cleaned images were then resized and preprocessed according to the input requirements of each model: 224 x 224 pixels for Vision Transformer (ViT) which uses patch-based embedding; 640 x 640 pixels for YOLOv8n and YOLOv11n which is consistent with their detection heads. In addition, we used background subtraction to isolate leaf regions and reduce the noise from the extraneous background. Pixel values were normalized to standard ranges that are appropriate for each architecture for training stabilization purposes. The dataset was then split into three subsets in 70:20:10 ratio, out of which 70% was used for training, 20% for validation and 10% was kept for final testing. Stratified sampling was done in order to balance the number of classes of plants - Healthy, Insect Pest, Leaf Spot and Mosaic Virus. This preprocessing pipeline guaranteed clean and consistent inputs to the models and robust evaluation results that could be compared.

Model 1: Vision Transformer (ViT)

The Vision Transformer (ViT) is a transformer-based deep learning architecture, in which the ideas of transformers originally introduced to natural language processing are applied to image classification. Instead of using convolutional kernels, ViT segments an input image into fixed-size patches of size 224x224 pixels, which are then flattened and projected to linear embeddings. In order to preserve the spatial information, a positional encoding is added and a special [class] token is added for the representation of the whole image. The embeddings of the patches are fed into a transformer encoder composed of multi-layers of multi-head self-attention and feed-forward networks, as well as normalization and residual connections for stabilizing the training. At the output, the [class] token embedding is passed through a multi-layer perceptron (MLP) head to predict the leaf as belonging to one of four classes: Healthy, Insect Pest, Leaf Spot or Mosaic Virus. As shown in Figure 3.2 this architecture helps ViT to capture the long-range dependencies and global contextual relationship information, which is suitable for making discriminations among subtle variations in leaf diseases.

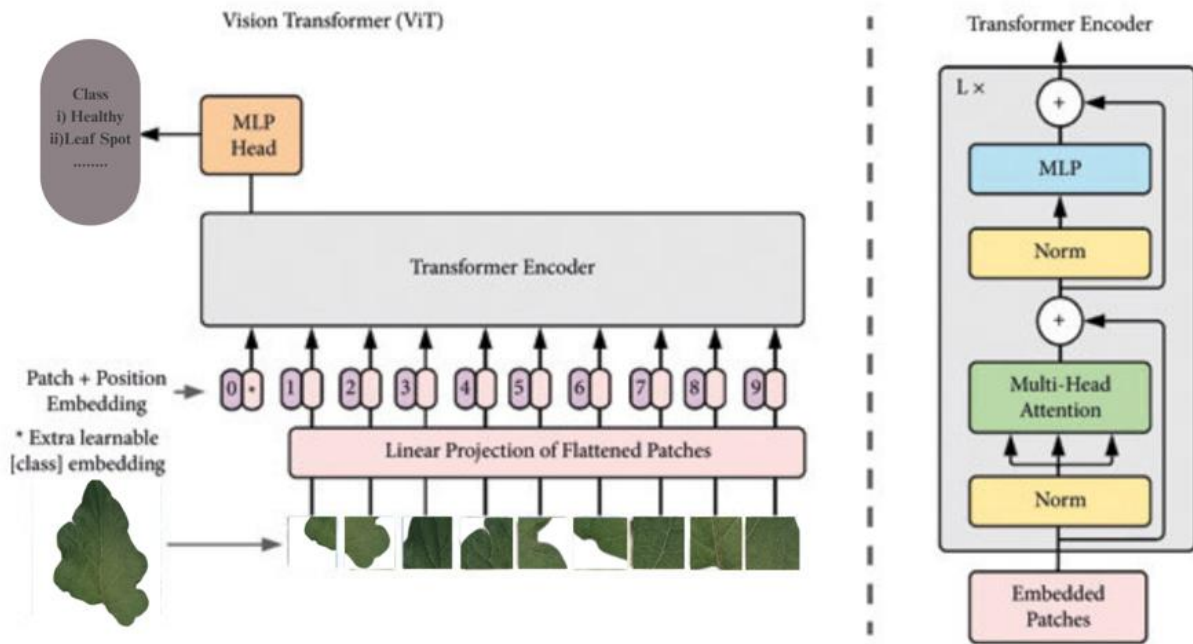


Figure 3.2: Vision Transformer (ViT) Model Architecture [31]

Model 2: YOLOv8n

YOLOv8n, created by Ultralytics, is an advanced one-stage detection model that is highly optimized for efficiency and speed. In this study, input image was resized to 640x640 pixels according to the model's requirement for input image resolution. The structure is in accordance with the standard backbone-neck-head structure. The backbone passes low- to high-level features through convolutional layers and can employ C2f modules that offer an efficient way to represent features while keeping the model lightweight. Multi-scale feature fusion is performed in the neck via concatenation and upsampling, to fuse information from multi-scale features for enhanced detection of both large and small features. The detection head directly provides predictions for class prediction, so the YOLOv8n is very efficient for real-time inference. Unlike previous versions of YOLO, YOLOv8 is anchor-free, which improves scale adaptability and reduces computational cost. As indicated in Figure 3.3 YOLOv8n Architecture, it has a lightweight structure that maintains an equilibrium between accuracy and inference speed. In this study, YOLOv8n was used for four-class classification of eggplant leaf images, which provided good and fast recognition suitable for field-level applications.

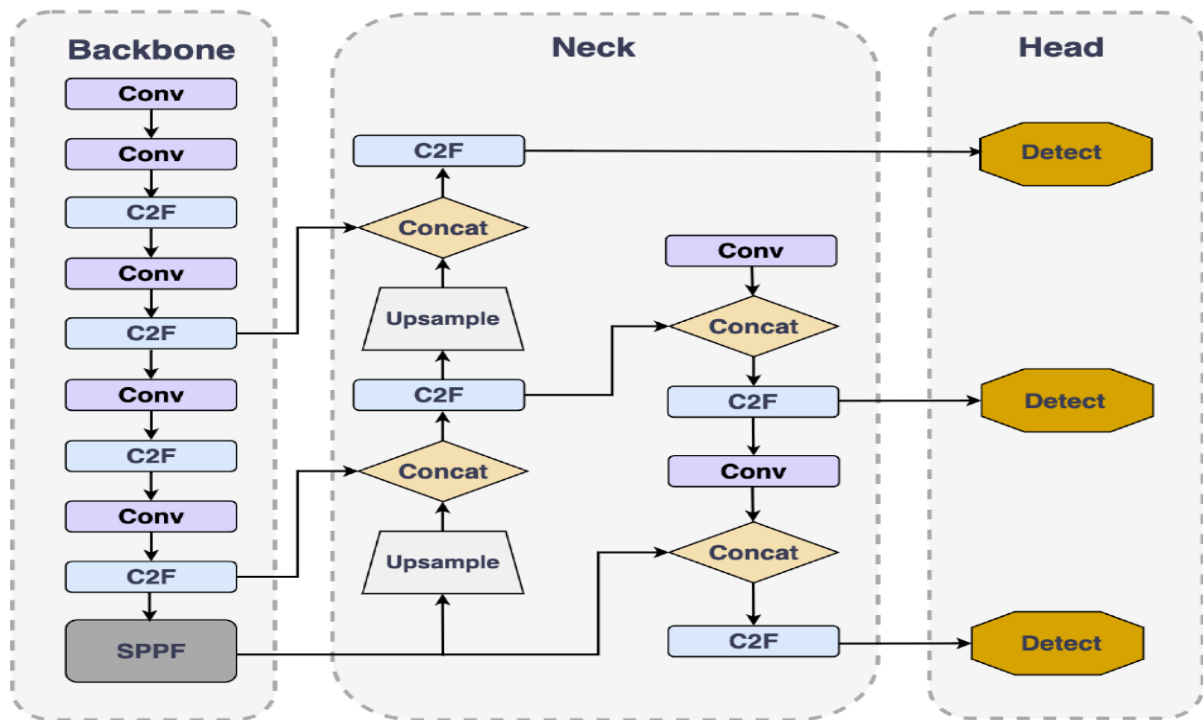


Figure 3.3: YOLOv8n Model Architecture [32]

Model 3: YOLOv11n

YOLOv11n architecture is an improved version of the YOLO family, and it is offering improve predictive performance, along with enhancement of segmentation effectiveness. Like YOLOv8, it works with input images of size 640x640 pixels, but brings a number of architectural improvements. Accordingly, as already depicted in Figure 3.4 YOLOv11n Architecture, the backbone is made up of C3K2 modules for quick feature extraction and SPPF (Spatial Pyramid Pooling - Fast) layers for multi-scale learning. For better capturing of subtle or overlapping features, a cross-scale feature fusion module C2PSA is used together with deeper concatenation paths in the neck to enhance feature fusion ability. Classification outputs are produced directly from the head, which is flexible enough to be used for segmentation if that is desired. Improved precision: YOLOv11 focuses on higher recall, reducing false negatives and ensuring that even small or subtle symptoms are not missed. This makes it particularly suitable for agricultural work where diseases can present themselves in complicated or overlapping patterns. In this study, in addition to comparing ViT and YOLOv8n, we benchmark YOLOv11n against, to determine whether the extra complexity of YOLOv11n is worth the measurable gain in accuracy and robustness for eggplant leaf disease detection.

configured. The models are all cooked with the same input specifications to make a fair comparison.

- VII. Training protocol: The hyperparameters are kept the same in the models being trained. Early termination and checkpointing help to avoid overfitting and promote reproducibility.
- VIII. Evaluation protocol: Performance is assessed on the basis of accuracy, precision, recall, F1-score and mean Average Precision (mAP). Such measures give a detailed idea of the classification and detection efficiency.
- IX. Operating-point calibration: Classification and detection thresholds are tuned to achieve the highest accuracy versus recall. This is done to make it practical when it is applied in real time.
- X. Comparative analysis: The findings are compared to determine trade-offs among accuracy, recall and computational efficiency. This discussion identifies the most appropriate architecture in the process of detecting eggplant disease.

3.4 Task Allocation

The table 3.1 shows the schedule of the major events during each month of the project, week 1 to week 33. The work was organized into simplified stages to guarantee the coherence and effective monitoring, with the final step of developing a web interface and hosting to deploy it.

Table 3.1: Task Allocation Timeline

Tasks	Weeks																
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33
Project setup, scoping, repo & seed																	
Data Collection (field + online)																	
Manual curation & Pre-processing																	
Train Models (ViT vs YOLOv8n vs YOLOv11n)																	

Chapter 4

Implementation and Results

The chapter reports model training and evaluation results and gives the experimental setup, including hardware and software environment. It gives a comparative analysis of the ViT, YOLOv8n and YOLOv11n and then discusses the advantages and disadvantages of the three.

4.1 Environment Setup

4.1.1 Hardware

This project needed a combination of local computing resources and cloud-based computing environments to implement it. I set up the local workstation with the AMD Ryzen 5600 six-core processor, 16 GB of RAM in the form of DDR5, and an NVIDIA GeForce RTX 3050 graphics card with 6 GB of GDDR6 memory. Preprocessing of datasets, preliminary experiments (small scale), and preliminary validation of models were done using this system. To support training and evaluation of a large scale, the project used Kaggle Notebook environments which made available two NVIDIA Tesla T4 GPUs, optimized to perform deep learning activities. The combination of hardware used did make certain that pre-testing and full scale training could be performed effectively.

4.1.2 Core software & frameworks

The implementation of the project was carried out using Python programming language and well-known deep learning and data processing libraries. PyTorch was used as the main deep learning framework, Torchvision was used for image transformations and Ultralytics YOLO for YOLOv8 and YOLOv11 model training. The experiments were performed with Vision Transformer (ViT) using the Transformers library from HuggingFace. In addition, OpenCV was used for image preprocessing, and scikit-learn was used for evaluation and performance analysis. All experimental code was implemented in Jupyter Notebook environments both on local and cloud platforms to allow for interactive testing, visualization and reproducibility of results.

4.2 Comparative Analysis

The experimental performance of the three selected models, namely, Vision Transformer (vit_basepatch16_224), YOLOv8n and YOLOv11n, are compared against each other using the curated eggplant leaf dataset. Each model was trained and tested in the same experimental conditions, and results were interpreted in terms of accuracy, precision, recall, F1-score and mean Average Precision (mAP). In addition to the simpler metrics, precision-recall curves, and confusion matrices were analyzed in order to get a deeper insight.

4.2.1 ViT (vit_base_patch16_224)

79.90% (204) Accuracy, 0.80 macro-Precision, 0.76 macro-Recall, 0.776 macro-F1. Per-class F1 (P/R/F1, support)-

Table 4.1: Test Results of ViT Model

Class	Precision	Recall	F1-score	Support
Healthy	0.8280	0.9390	0.8800	82
Insect_Pest	0.7692	0.6977	0.7317	43
Leaf_Spot	0.7292	0.7292	0.7292	48
Mosaic_Virus	0.8750	0.6774	0.7636	31
Accuracy			0.7990	204
Macro avg	0.8003	0.7608	0.7761	204
Weighted avg	0.7995	0.7990	0.7956	204

As shown in Table 4.1: Classification Report, ViT (vit_base_patch16_224) achieves an overall accuracy of 79.9% with weighted F1-score: 0.796 on the eggplant leaf disease dataset. Among the four classes, Healthy leaves presented the best performance with a recall of 0.939 (77 samples correctly identified out of 82) and we found a precision of 0.828, a consequence of false positives from samples misclassified from other classes. In contrast, Insect Pest and Leaf Spot showed moderate performance with F1-scores close to 0.73, showing that there are frequent misclassifications between them probably because of overlapping visual features. Mosaic Virus had the best precision (0.875) but relatively weak recall (0.677) which means there were some infected leaves that were falsely labeled as Healthy or Leaf Spot. The difference between macro and weighted averages also shows the class imbalance in the dataset. Overall, ViT shows good performance in classifying

Healthy leaves and diseased leaves, but it fails to balance recall between minority classes, which indicates that more balanced data and better augmentation methods are required.

4.2.2 YOLOv8n

Overall, YOLOv8n model showed good overall performance on eggplant leaf disease dataset with an average precision (mAP@0.5) of 0.874 with similar detection across classes.

Table 4.2: The testing result of YOLOv8n Model.

Class	Precision	Recall	F1-Score	mAP@0.5
Healthy	0.841	0.890	0.865	0.938
Insect Pest	0.689	0.860	0.765	0.871
Leaf Spot	0.801	0.670	0.730	0.807
Mosaic Virus	0.832	0.798	0.815	0.880
Overall	0.791	0.805	0.794	0.874

As shown in Table 4.2, YOLOv8n achieved promising overall performance with mAP@0.5=0.874, mAP@0.5-0.95=0.874, F1-score=0.794. Class-wise, it is found that the strongest class is Healthy leaves (F1 = 0.865, mAP = 0.938) whereas the lowest is Leaf Spot (F1 = 0.730, mAP = 0.807) because of low recall. Insect Pest achieved high recall and lower precision and Mosaic Virus had balanced detection (F1 = 0.815). Overall, YOLOv8n is an effective algorithm for practical eggplant disease detection with reliable results.

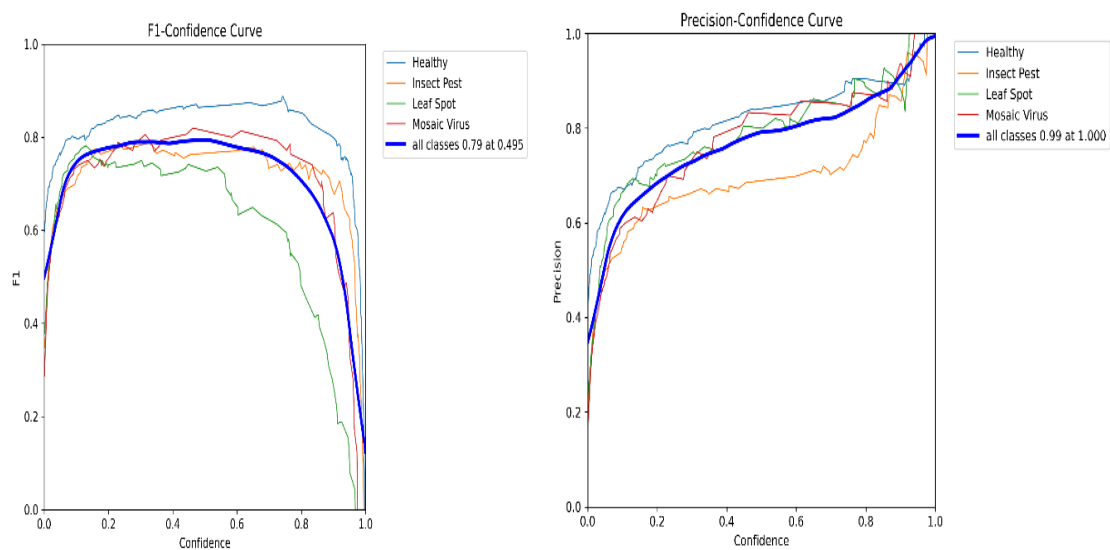


Figure 4.1: F1-Confidence Curve and Precision-Confidence Curve (YOLOv8n)

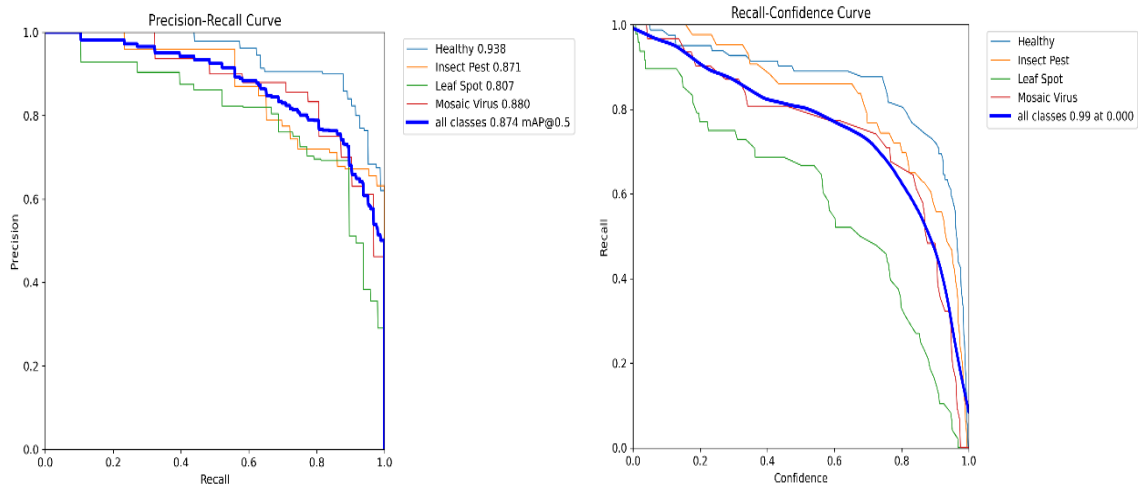


Figure 4.2: Precision-Recall Curve and Recall-Confidence Curve (YOLOv8n)

As can be seen in Figure 4.1 (F1-Confidence Curve and Precision-Confidence Curve) and 4.2 (Precision-Recall Curve and Recall-Confidence Curve), the stability of the performance curves is consistent with the level of confidence. There were good precision-recall trade-offs between the Healthy and Mosaic Virus classes, and Leaf Spot was outperformed by poorer recall values. The general curve validated the model $mAP\ 0.5 = 0.874$, which shows credibility at various thresholds.

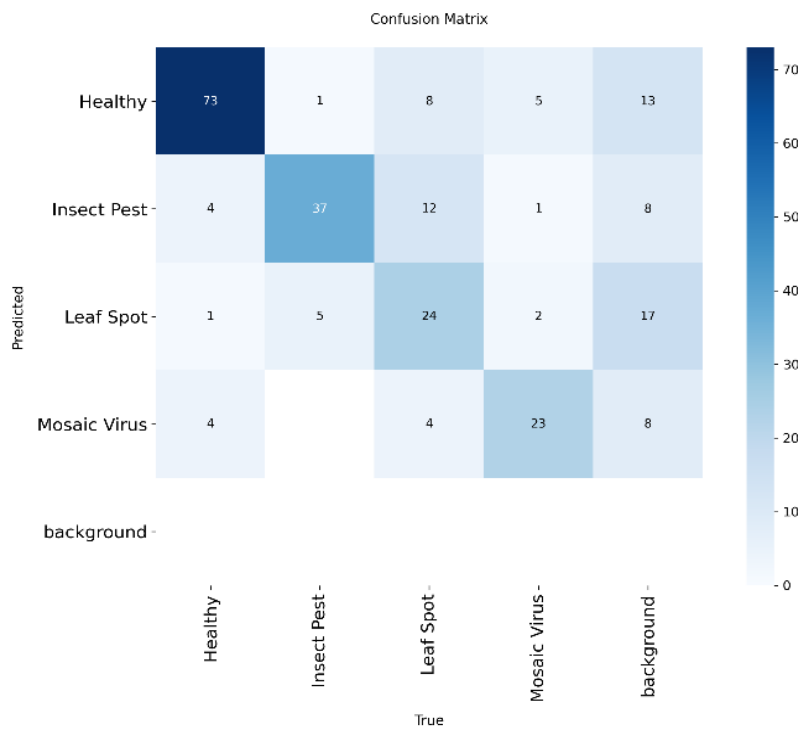


Figure 4.3: Confusion Matrix (YOLOv8n)



Figure 4.4: Confusion Matrix Normalized (YOLOv8n)

Both healthy (0.89) and insect pest (0.86) were classified with high accuracy by YOLOv8n as illustrated in Figure 4.3 and Figure 4.5. Mosaic Virus became moderately reliable (0.74) and Leaf Spot least reliable (0.50), frequently being confused with Mosaic Virus. This shows the strength of YOLOv8n in common classes but the weakness in distinguishing visually similar diseases.

4.2.3 YOLOv11n

The performance of the YOLOv11n model on the eggplant leaf disease dataset, in which the mAP@0.5 is 0.869 and the detection is uniform across the classes.

Table 4.3: Illustrating the testing result of YOLOv11n Model.

Class	Precision	Recall	F1-Score	mAP@0.5
Healthy	0.734	0.909	0.812	0.915
Insect Pest	0.665	0.860	0.750	0.866
Leaf Spot	0.704	0.841	0.767	0.801
Mosaic Virus	0.803	0.903	0.851	0.895
Overall	0.726	0.879	0.795	0.869

As Table 4.3 evidences, the overall performance of YOLOv11n was excellent with $mAP_{869} = 0.868 = 0.868 = 0.868$. Mosaic Virus was the strongest ($F1 = 0.851$), and Insect Pest was a little weaker ($F1 = 0.750$) because of worse precision. Healthy and Leaf spot were also reliably detected with a F1-score greater than 0.76 In general, YOLOv11n demonstrated better recall of all classes than YOLOv8n.

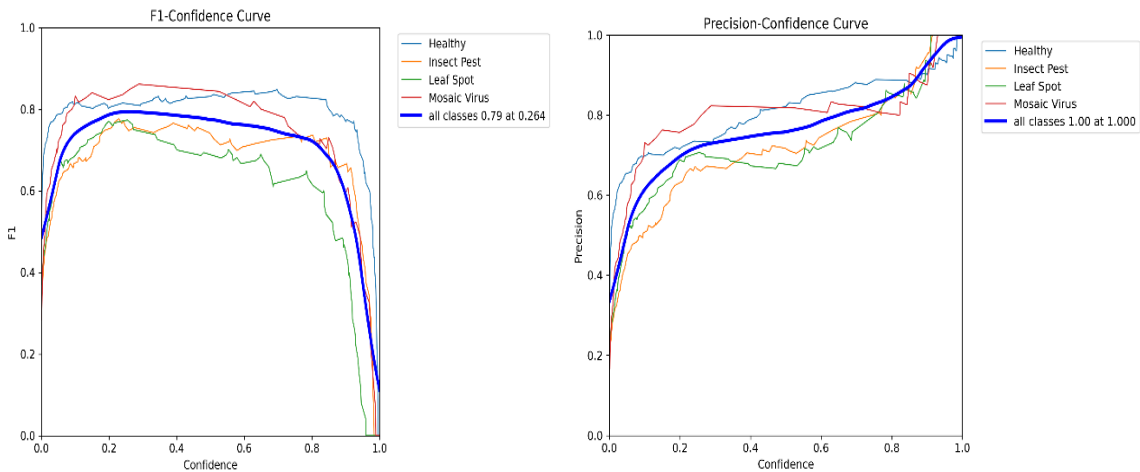


Figure 4.5: F1-Confidence Curve and Precision-Confidence Curve (YOLOv11n)

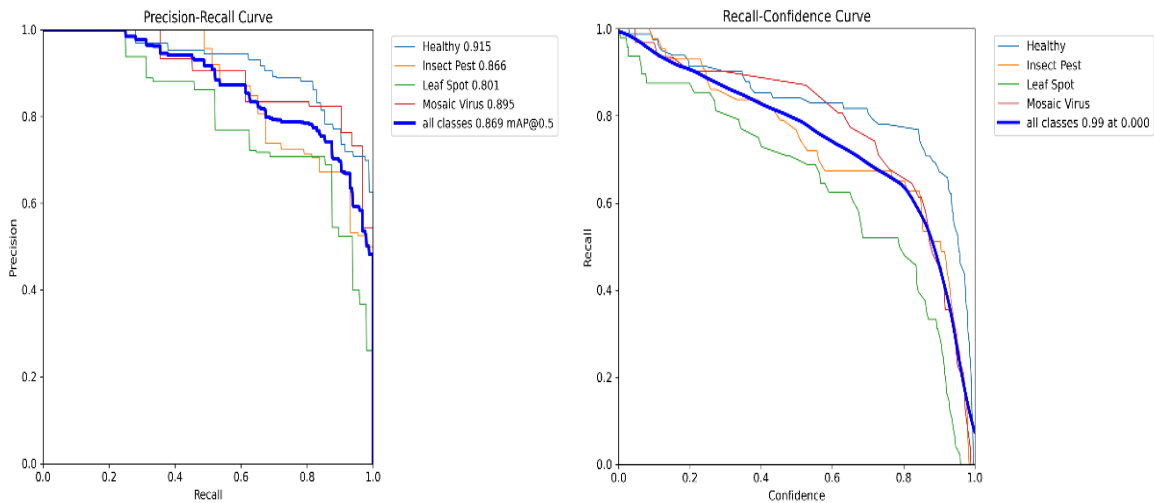


Figure 4.6: Precision-Recall Curve and Recall-Confidence Curve (YOLOv11n)

Figure 4.5 (F1-Confidence Curve and Precision-Confidence Curve) Figure 4.6 (Precision Recall Curve and Recall-Confidence Curve), indicate that the overall F1 score is 0.79 at the confidence value 0.264, with precision increasing steadily and reaching 1.0 with Healthy and Mosaic Virus, again, performing best and poorer respectively.

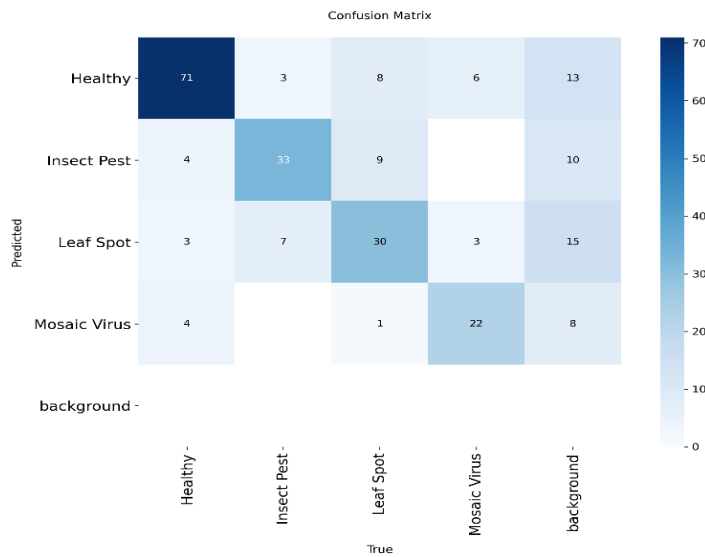


Figure 4.7: Confusion Matrix (YOLOv11n)



Figure 4.8: Confusion Matrix Normalized (YOLOv11n)

The confusion matrices (Figure 4.7 and Figure 4.8) depict how YOLOv11 performs the classification of the four disease classes. Healthy leaves are well recognized in the model, with a correct classification rate of 87 percent, although few are misclassified with Leaf Spot and Mosaic Virus. The accuracy of Insect Pest is approximately 77% and there is significant confusion with Leaf Spot. The weaker category is Leaf Spot, which has a correct prediction of only 62 percent, and is commonly confused with Mosaic Virus or Insect Pest. Mosaic Virus has a higher performer at 71, and also overlaps Leaf Spot. These findings indicate that overall performance of YOLOv11 is good, but inter-class similarities particularly between Leaf Spot and Mosaic Virus is a problem.

4.3 Results and Discussion

The effectiveness of the three models ViT, YOLOv8n, and YOLOv11n was measured by the standard measures: Precision, Recall, F1-Score, and mAP. The mathematical base of each measure is discussed and the results compared among the models.

- I. Precision (P): It is used to determine the fraction of positive predications that are actually true to all the positive predications, which should be interpreted as the ability of the model not to commit false positives. Precision is expressed in equation (i).

$$P = \frac{TP}{TP + FP} \quad (i)$$

With (i), ViT has achieved a total precision of 0.800, and it has performed best at the Mosaic Virus class with a precision of 0.875. YOLOv8n gave the next results: 0.791, doing very good in the Healthy leaves (0.841) but worse in the Insect Pest (0.689). YOLOv11n was the lowest of the three with a performance of 0.726, and was competitive with Mosaic Virus at 0.803. These findings show ViT is more precise but YOLOv8n is more balanced across classes.

- II. Recall (R): It measures the relative number of correct positive predictions in all positive samples (not false negative). The definition of the recall is equation (ii).

$$R = \frac{TP}{TP + FN} \quad (ii)$$

Based on (ii), ViT had 0.761 in the recall but it had a problem with Mosaic Virus (0.677). The total recall of YOLOv8n was 0.805 (with the highest Insect Pest class 0.860). YOLOv11n had the highest recall of 0.879 and this is very useful in agricultural settings where missing detections is a serious problem.

- III. F1-score: The F1-score gives a trade-off between the precision and the recall in terms of a harmonic mean. Its equation is in equation (iii).

$$F1 = \frac{2 \times P \times R}{P + R} \quad (iii)$$

Using (iii), ViT was 0.776, which is good in the Healthy class (0.880). YOLOv8n achieved a score of 0.794 with a high level of consistency between classes. YOLOv11n edged over with 0.795, which gains a lot of recall at the cost of precision. This shows the advantage of YOLOv11n to detect more cases of the disease, but at the cost of more false alarms than YOLOv8n.

IV. Mean Average Precision (AP): Mean Average Precision (mAP) is adopted to evaluate detection precision at a set of thresholds of Intersection over Union (IoU). The measure is given in equation (iv).

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (iv)$$

Where AP_i is the Average Precision of class i , and N is the number of classes. Using (iv), YOLOv8n achieved the highest values with $mAP@0.5 = 0.874$ and $mAP@0.5-0.95 = 0.874$. YOLOv11n followed closely at 0.869 and 0.868, respectively. ViT does not natively compute mAP as it is purely a classification model.

In summary, the analysis shows that ViT maintains high precision, but its lower recall limits its utility for real-world detection. YOLOv11n delivers the strongest recall, ensuring fewer diseased samples are missed, though its lower precision may introduce more false positives. YOLOv8n, on the other hand, provides the most balanced results, excelling in mAP and maintaining a solid trade-off between precision and recall. Therefore, YOLOv8n emerges as the most effective model overall for research purposes, while YOLOv11n offers complementary value in high-recall scenarios.

4.3.1 Implementation of the Eggplant Disease Detection System

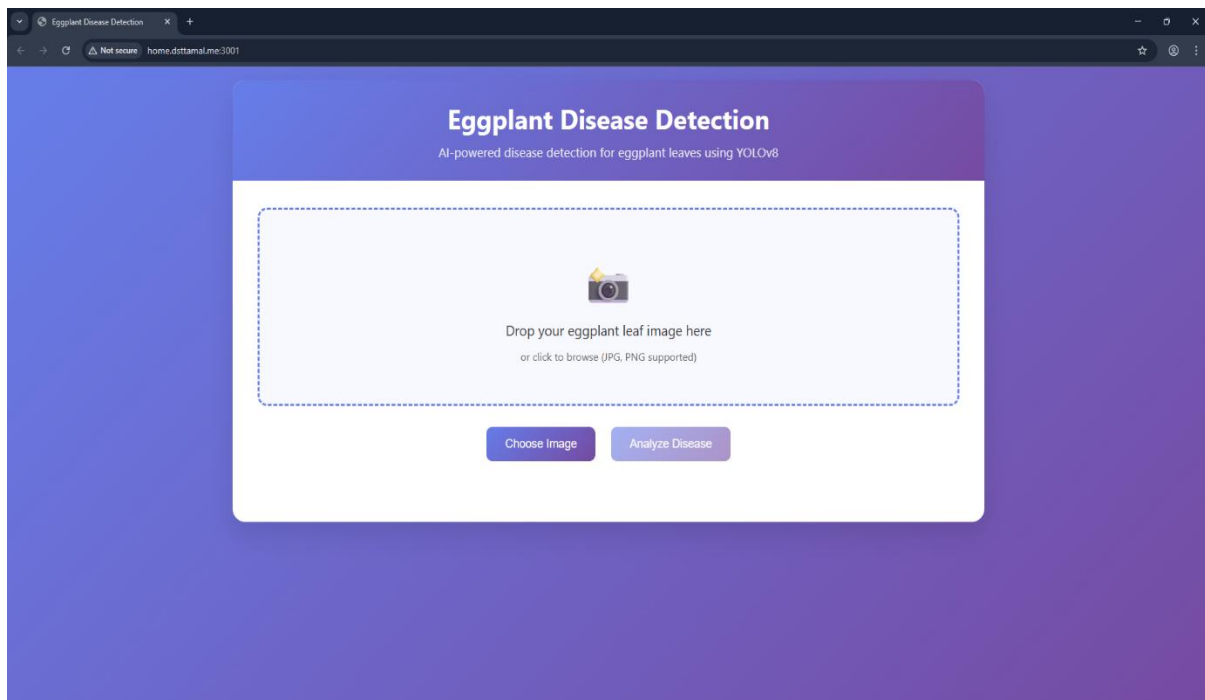


Figure 4.9: Image Upload Interface

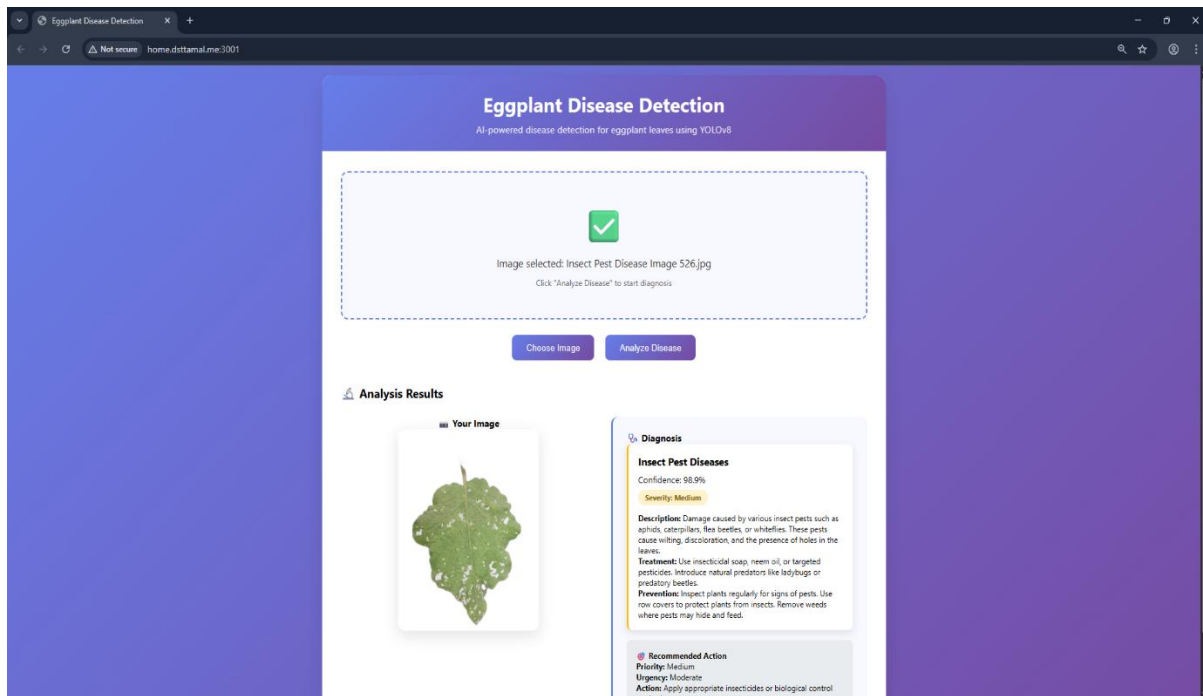


Figure 4.10: Disease Detection Results

The eggplant disease detection system resorts to the use of a user friendly Web interface where leaf images are uploaded. The interface is shown in the Figure 4.9 where in one can simply drag and drop or browse to choose an image to be analyzed. After uploading, the system identifies the diseases on the fly with the use of the YOLOv8n model. Figure 4.10 depicts the outcomes of the disease identification, with an insect pest having a confidence count of 98.9. The system contains information that is detailed such as the severity levels, treatment, and preventive measures. This methodology provides a viable, effective approach to real-time detection of agricultural disease.

4.4 Summary

In this chapter, I provided a comparative analysis of three models, ViT, YOLOv8n, and YOLOv11n, on the dataset of eggplant leaf disease. The analysis has taken into account several performance metrics, such as Precision (i), Recall (ii), F1-score (iii) and mAP (iv). ViT showed moderate precision of 79.9, which is appropriate to handle classification but not localization. YOLOv8n recorded the best mAP at 0.5 (0.874) with consistent F1, which indicates that it is a strong and effective detector. Compared to YOLOv11n, it had a better recall (0.879) and competitive mAP@0.5 was lower (0.869) which showed good sensitivity but low precision. Confidence-based curve and confusion matrix testing also supported the conclusion that YOLOv8n is the most reliable model in general, and YOLOv11n offered a different option when recall was

a priority. Combined, these results confirm the relevance of considering models on the basis of multiple measures as opposed to depending on one indicator. The major accomplishment of the study is that it implemented the YOLOv8n model in real-time via a web interface. The system enables users to post pictures and get immediate disease detection overwhelmings, as well as, treatment and prevention recommendation, which makes it a useful tool to farmers. This online feature illustrates the capabilities of AI-based disease detection systems to enable smart farming, enhance early detection, cut down on crop loss, and increase productivity.

Chapter 5

Engineering Standards and Design Challenges

This chapter talks of software, hardware and communication standards of the project. It also deals with societal impact, sustainability, project management and maps the work to complex engineering problems and activities.

5.1 Compliance with the Standards

The project is based on accepted software and hardware standards to achieve reliability, portability, and interoperability. The standards to be used are machine learning workflow, dataset management, embedded hardware specifications.

5.1.1 Software Standards

In the case of software, the project follows Python PEP 8 (coding style), ISO/IEC 25010 (software quality model) and ML experiment tracking best practices like versioning datasets and reproducibility. Other options such as Google Python Style Guide or less strict custom conventions may make coding easier but will risk inconsistency among teams. PEP 8 was selected as it strikes the right balance between readability and portability and is well-known in the field of ML research. Correspondingly, ISO/IEC 25010 provides that the implementation of the models can be assessed in terms of maintainability and reliability. Community support and reproducibility are also ensured by the use of standardized ML frameworks (PyTorch, Ultralytics YOLO).

5.1.2 Hardware Standards

Compliance on the hardware front can be IEEE 802.11 (IoT connectivity) and IEEE 754 (floating-point operations, necessary in GPUs/TPUs) and consistency in data representation with ISO/IEC 2382 (information technology vocabulary). Other options, such as custom hardware accelerators (FPGAs) or vendor-specific floating-point software might be faster, but at the expense of compatibility and portability. The selection of CUDA-enabled GPUs should provide a balance between calculation performance, support

of PyTorch/YOLO models, and availability in the cloud (e.g., Kaggle, Google Colab).

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The eggplant leaf disease detection system can greatly enhance the lives of the farmers especially in the developing area where crop diseases can cause a huge financial loss. Early and correct diagnosis of diseases like Leaf Spot, Mosaic Virus and Insect Pest will significantly help in reducing losses in crop yielding as farmers are able to ensure timely intervention measures are taken e.g., through application of specific treatment. This contributes to not only protecting their livelihood, but also fosters a more sustainable method of farming since fewer broad-spectrum pesticides have to be used. The system enables farmers to make effective decisions through a reliable, user-friendly tool in detecting the disease, resulting in improved crop management and higher output. Moreover, it has the effect of creating confidence in agriculture, thereby motivating the younger generations to consider farming as a good career choice hence the survival of agriculture in the countryside. Such AI-based solutions can further lift whole communities and improve food security, economic conditions, and health in the long-term.

5.2.2 Impact on Society & Environment

The social impact of this project is much wider than that of individual farmers. It will help in curbing the excessive application of pesticides as a key environmental issue, simply because early detection of disease will be realized. Not only pesticides poisoning beneficial insect species and pollinating insect species coupled with contaminating soil and water, overuse of pesticides impacts biodiversity and the health of the population. This system can alleviate those concerns by encouraging more precise use of pesticides, and hence reducing adverse effects on the environment. At the larger scale of society, it helps to provide the stable and sustainable supply of food. Since agriculture is a central activity in most developing countries, enhancing crop production and minimizing wastes directly leads to food security, which is central in (a) boosting economic growth and (b) eliminating poverty. The project promotes green farming as well as making the farming processes more productive, hence long-term sustainability. The system is a necessary precursor to a technologically responsible and ecologically responsible form of agriculture, which will be made possible by the adoption of AI technologies.

5.2.3 Ethical Aspects

When applied to the context of small-scale farmers, the application and implementation of AI-powered systems have an ethical component at their core. The ethical integrity of this project is assured through fairness and transparency during the lifecycle of the project. The models were trained with a dataset that was not inherently biased and we ensured that the dataset is diverse enough to capture variability in environmental conditions and plant health. Moreover, the decisions made by the model can be easily interpreted, and farmers can have a belief in the system. Another important consideration is data privacy and ownership, where all data obtained is not used commercially to exploit the situation but to facilitate disease detection. Moreover, by delivering a product that is affordable to smallholder farmers, we are helping to close the technological gap and enhance the fair use of AI in agriculture. The ethical objective of the given project is to make the advantages of AI technologies available to all interested parties, particularly, to those who can enjoy the benefits of such advances the most.

5.2.4 Sustainability Plan

This project is sustainable because it is long-term usable, has an impact on the environment, and is economically viable. The employed models, especially YOLOv8n, are efficient, and it is expected that the system can be executed on a smartphone or other inexpensive hardware and will be accessible by many farmers in limited-resource environments. Open-source frameworks are also given a high priority in the project and the fact that it can always be updated and improved with little cost also means that it will be sustainable without the need to pay continuous licensing charges. At the environmental front, this decrease in the use of pesticides by proper detection of diseases will be in tandem with sustainable agriculture which will help in protecting the ecosystem as well as avoiding soil and water pollution. The economic perspective of the system is that it saves farmers the financial burden linked to crop failures due to undiagnosed diseases, giving the farmers a cost-efficient approach to improving their yield. Moreover, the system can be expanded later to mobile applications, which will further increase the accessibility of the system, making the project more scalable in the long term. This integrated approach guarantees the economic, environmental and social sustainability of the project in the long term.

5.3 Project Management and Financial Analysis

This project followed an Agile-based management style in terms of weekly sprints and biannual supervisor meetings to monitor timely progress and focus on research goals. The tasks were broken into distinct stages - beginning with project setup and data acquisition and proceeding to model training, comparative evaluation and deployment. This repetitive cycle provided flexibility in tackling issues like dataset quality and computational overhead and ensuring that deliverables were met on time.

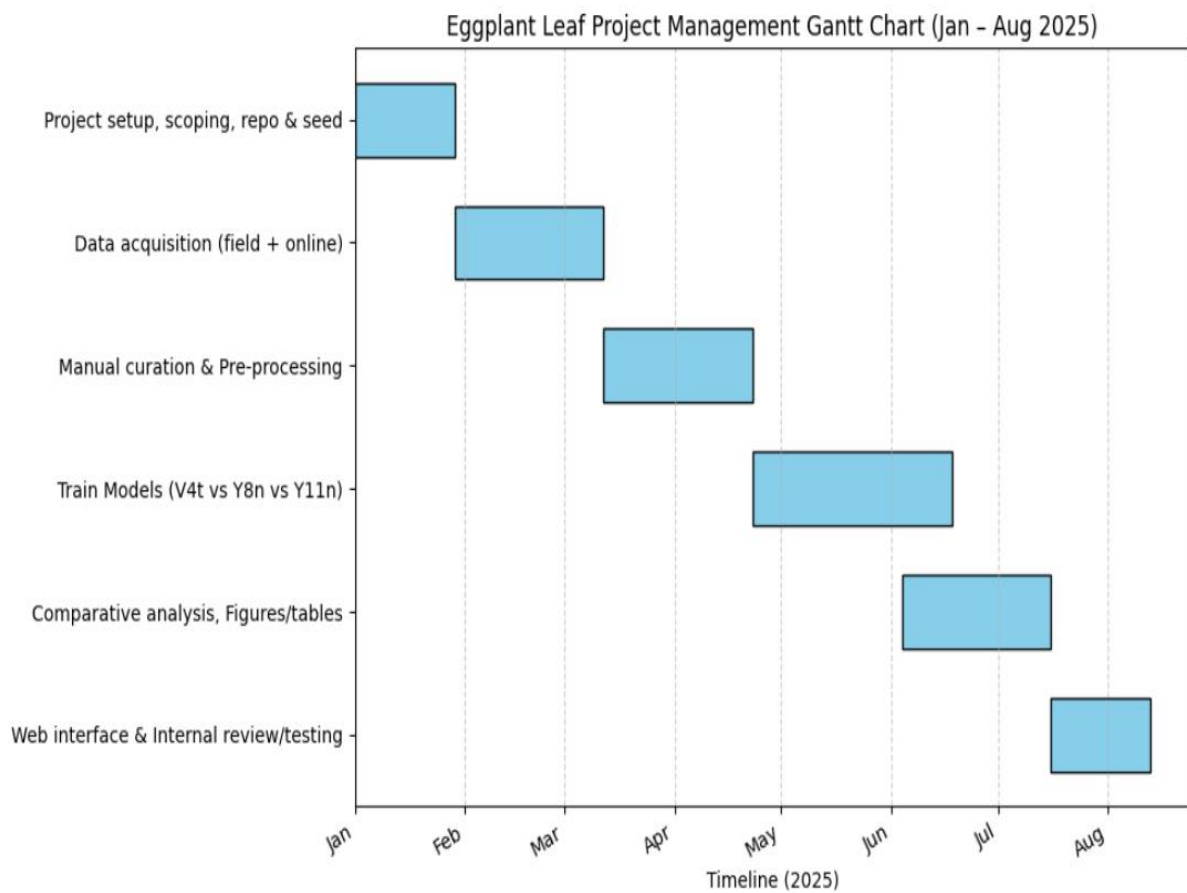


Figure 5.1: Project Management Gantt Chart

The Gantt chart below, Figure 5.1, shows the project timeline, between January and August 2025. These stages consist of setting up the project, field and online data collection, manual curation and preprocessing, model training (ViT, YOLOv8n, YOLOv11n), comparative analysis with visualizations, and the creation of a web interface, as well as an internal review and testing. This was a planned timetable, which made sure that both the experimentation and verification went without any hitches as major milestones were reached in a well organised manner.

Table 5.1: Financial Analysis table.

SN	Item or Component's	Cost in BDT
01	Computing Infrastructure: AMD Ryzen 5 5600 (6-core), 16 GB DDR5 RAM, NVIDIA RTX 3050 (6 GB), Kaggle Notebook (T4×2 GPU runtime), storage devices, and other peripherals	75,300
02	Power consumption for GPU computing	10,000
03	Field Data Collection & Transport: Travel expenses, logistics, and field setup	7,000
04	Web Domain & Hosting	12,600
05	Miscellaneous: Printing, report preparation, contingency costs	3,000
Total Cost		101,900 BTD

Other key features of the project were financial planning. Expenses were spread over hardware and software assets, field work, web hosting and overheads. Computing infrastructure with AMD Ryzen 5 5600 processor, 16 GB DDR5 RAM, NVIDIA RTX 3050 graphics card and Kaggle GPU runtime received the biggest budget. This system gave the computational capabilities required to achieve effective deep learning training. Other expenses were incurred in power consumption, field data collection (including transport) and implementation of a web-based interface to internal testing. Other small costs were also taken into account like printing and report preparation. Table 5.1 shows the final estimated budget and reflects a balanced distribution of resources between technical and operational requirements. Majority of the investments were warranted because they directly supported the model training and evaluation processes, which is computationally intensive.

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

Applying ViT, YOLOv8n, and YOLOv11n as an eggplant disease detection system may be considered a complicated engineering issue since it demands multidisciplinary viewpoints, sophisticated methods of computation, and logical decision-making within conditions. Mapping with problem solving categories is

required in this section the 5.2 table.

Table 5.2: Mapping with Complex Engineering Problem.

EP1 Dept of Knowledge	EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stakeholder Involvement	EP7 Interdependence
✓		✓	✓			✓

EP1: In this project, deep learning architectures (ViT, YOLOv8n, YOLOv11n) had to be integrated with domain-specific agricultural knowledge regarding the disease of the eggplant. We did this by creating experiments that integrated modern computational techniques like patch embeddings in ViT with the real-time detection of YOLO models. Also, the trainees trained with GPUs on Kaggle T4x2 environments showed functional experience of resource optimization. This level of interdisciplinary understanding is obviously an indication of how the project complies with EP1.

EP3: The project passed EP3 by undergoing strict comparative analysis between models. We did not examine accuracy measures; rather, we examined precision, recall, F1-score, confusion matrices, and mAP figures. Evaluations involving curves including Precision-Recall curves, F1- Confidence curves and Recall- Confidence curves also served to emphasize the model behavior at different thresholds. This holistic methodology ensured that analysis was not mere superficial analysis and therefore met the EP3 criteria.

EP4: Issues such as imbalance of classes, noise in the data set and overfitting are common in machine learning. In this project, however, these problems have been applied to a new context, the agricultural image data. To solve the latter, we used background removal, resizing of the image, and judicious dataset division (70-20-10 ratio), with equal evaluation. These extrapolations of familiar problems to a new system were direct accomplishments of EP4.

EP7: Interdependence in our project was in hardware, software and domain knowledge. High-performance training was made possible with hardware components including GPUs and local systems. The computational backbone was based on software frameworks such as PyTorch, Ultralytics YOLO and Hugging Face ViT. In the meantime, the knowledge of the domain in plant pathology played a vital role in labeling and validating

the classes of disease. It is only with the collaboration of these mutually supporting components that the project was able to meet its goals, thus, fulfilling EP7.

Mapping with Knowledge Profile

Table 5.3: Mapping with knowledge Profile.

K1	K2	K3	K4	K5	K6	K7	K8
Natural Science	Mathematics	Engineering Fundamentals	Specialist Knowledge	Engineering Design	Engineering Practice	Comprehension	Research Literature
		✓	✓	✓	✓		✓

K3: This project relies on the major principles of engineering like linear algebra and probability theory and optimization which play a significant role in deep learning. Considering the above, the training of neural networks relies on the backpropagation algorithm of model training, gradient descent of optimization, and loss minimization algorithms. Both Vision Transformer (ViT) and YOLO architectures rely on these basic principles to undergo training and enable our system to classify plant diseases with high performance at a reasonable computation cost.

K4: This project involved a professional understanding in computer vision and artificial intelligence (AI). We used the Vision Transformer (ViT) and YOLO-based models (YOLOv8n, YOLOv11n), the latter of which requires an in-depth knowledge of patch embeddings, transformer encoders, convolutional backbones, and anchor-free detection. This information helped us develop models that perform well in classifying diseases in agricultural images. The incorporation of these modern technologies revealed the potential of expert knowledge to act as an innovator and improve performance in a difficult problem-solving situation, like detecting plant diseases.

K5: The design of the project was to have a systematic pipeline which started with collecting the data, then necessary preprocessing processes were followed like resizing of the image and removal of the background. A 70 20 10 split was then used to get the training set, the validation set and the test set. ViT and YOLO architectures were then applied to conduct multi-model evaluation. Their cautious design allowed making the solution both robust and versatile to the agricultural background so that the disease could be detected among various environmental conditions. The structured design of the

methodology is the most important part of the efficiency of the project.

K6: The project used the industry-standard engineering practices with modern tools and frameworks. Deep learning models were implemented in PyTorch and training and fine-tuning of the Vision Transformer (ViT) were achieved through Hugging Face Transformers. Ultralytics YOLO offered an effective detection system, and we used our CPU resources on both Kaggle T4x2 and local NVIDIA RTX 3050 computing resources to ensure computational efficiency. These tools made the project feasible and scalable so that the models could be used in a real-world agricultural environment with a small amount of resources.

K8: We have based our approach on a thorough examination of 30 research articles which have helped identify what is known about the methodologies and which gaps still exist in the literature. This study has shown the untapped potential of transformer-based architectures in agricultural uses, especially detection of eggplant disease. The synthesis of the results provided by others helped us define the strong and weak aspects of the existing models and choose a benchmarking model that combines the old and new methods. This guaranteed our approach was not only innovative but keeps up to date with the recent developments in the sphere of AI and agriculture.

5.4.2 Engineering Activities

The explored project could be summarised as provided below as applied to the framework of the Complex Engineering Activities and each of the below provided activities is further supported by the report findings.

Mapping with Complex Engineering Activities

Table 5.4: Mapping with Complex Engineering Activities.

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

EA1: The project made use of a wide variety of computing resources, such as a local Ryzen 5600 CPU and RTX 3050 GPU, and cloud-based Kaggle T4x2 GPUs to train and test

models. The sample was based on the images collected in the field and published in the repositories (Mendeley), which would represent the real world conditions in a balanced manner. Furthermore, deep learning with PyTorch, ViT with Hugging Face Transformers, and detection with Ultralytics YOLO were applied. This extensive collection of resources both enabled the project to address the issue in multiple ways and guaranteed its flexibility across environments.

EA2: This project was a success because of the interplay between different disciplines: software (deep learning architectures), hardware (GPU accelerators), domain knowledge (plant pathology) and the ultimate customers (farmers). Teamwork among these components played a vital role in the successful data gathering, data preprocessing, and deployment of the model. The model would not have been effective without active participation of the plant pathologists who would have ensured the correct labelling of the disease. Likewise, farmer feedback at the validation stage was used to verify the practicality and usefulness of the system in actual farming conditions, a full feedback loop.

EA4: The project is of great benefit to the society and the environment. Early identification of the disease will enhance protection of crops hence higher yield and income hence improved food security among rural populations. On the environmental front the system assists in saving the environment by reducing the excessive usage of pesticides which cause soil and water contamination. These advantages have a direct impact on the achievement of sustainable agriculture as they promote the more effective use of resources. The project provides a viable answer to facilitate economic development as well as environmental conservation in the agricultural sector.

5.5 Summary

This chapter has presented an excellent case study of the engineering underpinnings, engineering standards, and overall implication of the eggplant leaf disease detection project. The project was well aligned on internationally accepted standards of software and hardware and the application of high-tech software and hardware such as PyTorch, YOLO, and Vision Transformers can be justified in the context of ethics and sustainability. It also emphasized the benefits of the society and the environment, and showed that proper detection of the disease could mean a lot in the domain of food security, pesticide usage and sustainable farming. With the help of the system, farmers will

experience healthier crops and improved income as they will be able to detect diseases early and with high accuracy. The management of the project was designed in a way that it planned its finances and use of a Gantt chart to ensure the project was not only cheap but also completed on time. It was cross-tabulated with complex problem-solving standards (EP1, 3, 4, 7) and knowledge profile type (K3, K4, K5, K6, K8), showing the level of analysis and interdisciplinary knowledge needed. Also, it was linked to the types of engineering activities (EA1, EA2, EA4), which depict the multidimensional nature of the project and its engagement of different stakeholders, such as software developers, plant pathologists, and farmers. These aspects all indicated the strength, feasibility and stability of the study and made it applicable to both scholarship and practical farming.

Chapter 6

Conclusion

This chapter provides a summary of the research findings that focus on the comparative performance of ViT and YOLO models in detecting eggplant disease. It also points out the weaknesses of the present work and indicates future research directions.

6.1 Summary

This study was aimed at developing and comparatively analyzing the advanced models of deep learning to detect diseases in eggplant leaves, which is one of the most vulnerable crops to diseases like Leaf Spot, Mosaic Virus, Insect Pest. The research was inspired by the fact that real-time, accurate and scalable solutions to early detection of diseases were necessary in order to help in the reduction of crop losses, enhancement of farmer income and the sustainability of farming activities. To do this, a curated corpus of 2,032 images was composed of 1,008 field-collected samples and images retrieved publicly through Mendeley. All pictures were preprocessed (background removal, resizing, and normalizing) and the dataset was split into training (70%), validation (20%), and testing (10%) categories so that the evaluation was balanced. Three models under consideration were: Vision Transformer (ViT), YOLOv8n, YOLOv11n that all finished training under the same conditions with both local (RTX 3050) and cloud GPUs (Kaggle T4x2). Measures of evaluation were Precision, Recall and 3 F1-score and mean Average Precision (mAP), and visual evaluation such as Confusion matrices, and confidence curves. ViT was also very precise, but with low recall on minority classes and not able to detect objects in real-time. YOLOv11n had high recall and guaranteed fewer misdetections, but due to poor precision, it has a low overall reliability. On the contrary, YOLOv8n appeared to be the most balanced and obtained higher precision and recall, stable F1-score, and high mAP, in addition to providing real-time detection which can be used in practice. Such accuracy and efficiency allow YOLOv8n to be especially effective in the real-world agricultural setting, where farmers can rely on such a tool to detect diseases early. The research work presented here forms the basis of future improvements, such as adding additional environmental factors, introducing low-weight and mobile-friendly versions of transformers, deploying explainable AI (XAI) solutions such as Grad-CAM as the model

transparency tool, and building offline mobile or IoT applications. By working on all these future directions, the system will also be able to enhance the decisions made by farmers, enhance adaptability in all regions and extend to identify more diseases in crops. To sum up, this study shows how AI-based solutions have the potential to revolutionize modern agriculture. Directly underpinning the existing crop control methods, preventing pesticide abuse, and improving food security, it facilitates early and precise disease detection. The methodologies, models, and deployment strategies, created during this work, are scalable to sustainable, precision agriculture in the long run, so that both smallholder and commercial farmers can reap the benefits of AI technologies today and in the future.

6.2 Limitation

1. There is still limited diversity in datasets across regions, seasons, and environmental conditions, which may have an impact on generalization.
2. Though checked, not annotated, and possibly inter-class ambiguous (e.g. Leaf Spot vs Mosaic).
3. One of the weaknesses of this data is the unequal distribution of classes. In particular, samples of Healthy class are considerably larger than those of Mosaic Virus. This imbalance can make the model biased towards majorities classes and hence reducing its generalization power across minorities classes.
4. There are chances the model integrated into the website will make false predictions of the diseases when the input images are not within the domain of Eggplant leaf.

6.3 Future Work

1. Increase maximum size of dataset; multi-location multi-season data; expert-labeled and synthetic augmentation strategies.
2. Use better Transformer variations or hybrid YOLO-transformer models; use AutoML to fine-tune.
3. Combine model confidence output (SHAP, Grad-CAM) and explainability.
4. Implement agronomic recommendations and early-notification system as mobile/web app with offline inference; combine agronomic recommendations and early-alert system.

References

- [1] Howlader, S., Ahamed, M. S., Mojumdar, M. U., Noori, S. R. H., Siddiquee, S. M. T., & Chakraborty, N. R. (2025). A comprehensive image dataset for the identification of eggplant leaf diseases and computer vision applications. *Data in Brief*, *59*, 111353.
- [2] Liu, J., & Wang, X. (2025). EggplantDet: An efficient lightweight model for eggplant disease detection. *Alexandria Engineering Journal*, *115*, 308-323.
- [3] Paymode, A. S., & Malode, V. B. (2022). Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG. *Artificial Intelligence in Agriculture*, *6*, 23-33.
- [4] Abisha, S., & Jayasree, T. (2022). Application of image processing techniques and artificial neural network for detection of diseases on brinjal leaf. *IETE Journal of research*, *68*(3), 2246-2258.
- [5] KURSUN, R., & KOKLU, M. (2025). Classification of Eggplant Diseases Using Feature Extraction with AlexNet and Random Forest.
- [6] Islam, M., Ria, N. J., Ani, J. F., Masum, A. K. M., Abujar, S., & Hossain, S. A. (2022, January). Deep learning based classification system for recognizing local spinach. In *Advances in Deep Learning, Artificial Intelligence and Robotics: Proceedings of the 2nd International Conference on Deep Learning, Artificial Intelligence and Robotics, (ICDLAIR) 2020* (pp. 1-14). Cham: Springer International Publishing.
- [7] Haque, M. R., & Sohel, F. (2022). Deep network with score level fusion and inference-based transfer learning to recognize leaf blight and fruit rot diseases of eggplant. *Agriculture*, *12*(8), 1160.
- [8] Orchi, H., Sadik, M., Khaldoun, M., & Sabir, E. (2023). Automation of crop disease detection through conventional machine learning and deep transfer learning approaches. *Agriculture*, *13*(2), 352.
- [9] Tan, L., Lu, J., & Jiang, H. (2021). Tomato leaf diseases classification based

on leaf images: a comparison between classical machine learning and deep learning methods. *AgriEngineering*, 3(3), 542-558.

- [10] Ngugi, H. N., Akinyelu, A. A., & Ezugwu, A. E. (2024). Machine learning and deep learning for crop disease diagnosis: Performance analysis and review. *Agronomy*, 14(12), 3001.
- [11] Ajra, H., Nahar, M. K., Sarkar, L., & Islam, M. S. (2020, December). Disease detection of plant leaf using image processing and cnn with preventive measures. In *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)* (pp. 1-6). IEEE.
- [12] Xie, C., & He, Y. (2016). Spectrum and image texture features analysis for early blight disease detection on eggplant leaves. *Sensors*, 16(5), 676.
- [13] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and electronics in agriculture*, 145, 311-318.
- [14] Nahiduzzaman, M., Chowdhury, M. E., Salam, A., Nahid, E., Ahmed, F., Al-Emadi, N., ... & Haider, J. (2023). Explainable deep learning model for automatic mulberry leaf disease classification. *Frontiers in Plant Science*, 14, 1175515.
- [15] Bonkra, A., Bhatt, P. K., Rosak-Szyrocka, J., Muduli, K., Pilař, L., Kaur, A., ... & Rana, A. K. (2023). Apple leaf disease detection using collaborative ml/dl and artificial intelligence methods: Scientometric analysis. *International journal of environmental research and public health*, 20(4), 3222.
- [16] Wei, T., Chen, Z., Yu, X., Chapman, S., Melloy, P., & Huang, Z. (2024). Plantseg: A large-scale in-the-wild dataset for plant disease segmentation. *arXiv preprint arXiv:2409.04038*.
- [17] Wang, X., Yan, F., Li, B., Yu, B., Zhou, X., Tang, X., ... & Lv, C. (2025). A Multimodal Data Fusion and Embedding Attention Mechanism-Based Method for Eggplant Disease Detection. *Plants*, 14(5), 786.
- [18] Jayanthi, M. G., & Shashikumar, D. R. Eggplant leaf disease detection and segmentation using adaptively regularized multi Kernel-Based FuzzyC-

Means and Optimal PNN classifier.

- [19] Abisha, S., Mutawa, A. M., Murugappan, M., & Krishnan, S. (2023). Brinjal leaf diseases detection based on discrete Shearlet transform and Deep Convolutional Neural Network. *Plos one*, *18*(4), e0284021.
- [20] Xie, Z., Li, C., Yang, Z., Zhang, Z., Jiang, J., & Guo, H. (2024). YOLOv5s-BiPCNeXt, a Lightweight Model for Detecting Disease in Eggplant Leaves. *Plants*, *13*(16), 2303.
- [21] Zhang, Y., Zhang, D., Zhang, Y., Cheng, F., Zhao, X., Wang, M., & Fan, X. (2024). Early detection of verticillium wilt in eggplant leaves by fusing five image channels: a deep learning approach. *Plant Methods*, *20*(1), 173.
- [22] Krishnaswamy Rangarajan, A., & Purushothaman, R. (2020). Disease classification in eggplant using pre-trained VGG16 and MSVM. *Scientific reports*, *10*(1), 2322.
- [23] Kaniyassery, A., Goyal, A., Thorat, S. A., Rao, M. R., Chandrashekar, H. K., Murali, T. S., & Muthusamy, A. (2024). Association of meteorological variables with leaf spot and fruit rot disease incidence in eggplant and YOLOv8-based disease classification. *Ecological Informatics*, *83*, 102809.
- [24] Cemek, B., Tasan, S., Canturk, A., Tasan, M., & Simsek, H. (2023). Machine learning techniques in estimation of eggplant crop evapotranspiration. *Applied Water Science*, *13*(6), 136.
- [25] Demilie, W. B. (2024). Plant disease detection and classification techniques: a comparative study of the performances. *Journal of Big Data*, *11*(1), 5.
- [26] Sapkota, R., Ahmed, D., & Karkee, M. (2024). Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments. *Artificial Intelligence in Agriculture*, *13*, 84-99.
- [27] Hasan, A., Mumtaz, R., Mahmood, Z., Fayyaz, M., & Naeem, M. K. (2024). Wheat leaf localization and segmentation for yellow rust disease detection in complex natural backgrounds. *Alexandria Engineering Journal*, *107*, 786-798.
- [28] Haider, A., Arsalan, M., Hong, J. S., Sultan, H., Ullah, N., & Park, K. R.

- (2024). Multi-scale and multi-receptive field-based feature fusion for robust segmentation of plant disease and fruit using agricultural images. *Applied Soft Computing*, 167, 112300.
- [29] Islam, M. S., Sultana, S., Farid, F. A., Islam, M. N., Rashid, M., Bari, B. S., ... & Husen, M. N. (2022). Multimodal hybrid deep learning approach to detect tomato leaf disease using attention based dilated convolution feature extractor with logistic regression classification. *Sensors*, 22(16), 6079.
- [30] Islam, A., Raisa, S. R. S., & Khan, N. H. (2023). Enhanced leafy vegetable analysis: image classification and disease instance segmentation using deep learning techniques. *Available at SSRN 4470131*.
- [31] Prashanthi, B., Krishna, A. P., & Rao, C. M. (2025). LEViT-Leaf Disease identification and classification using an enhanced Vision transformers (ViT) model. *Multimedia Tools and Applications*, 84(21), 23313-23344.
- [32] Yao, G., Zhu, S., Zhang, L., & Qi, M. (2024). Hp-yolov8: high-precision small object detection algorithm for remote sensing images. *Sensors*, 24(15), 4858.
- [33] Khan, A. T., & Jensen, S. M. (2025). LEAF-Net: A unified framework for leaf extraction and analysis in multi-crop phenotyping using YOLOv11. *Agriculture*, 15(2), 196.

Plagiarism Report

ORIGINALITY REPORT

15%

SIMILARITY INDEX

10%

INTERNET SOURCES

8%

PUBLICATIONS

8%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	5%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
3	Submitted to United International University Student Paper	1%
4	arxiv.org Internet Source	<1%
5	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dharendra Kumar Shukla. "Recent Trends in Intelligent Computing and Communication", CRC Press, 2025 Publication	<1%
6	Shankar Babu, Mahesh Babu Kota. "Synergies in Smart and Virtual Systems using computational intelligence", CRC Press, 2025 Publication	<1%
7	www.pubs2.ascee.org Internet Source	<1%
8	Ibrar Hussain, Rifaqat Ali. "Robust Leaf Disease Detection Using Complex Fuzzy Sets and HSV-Based Color Segmentation Techniques", Acadlore Transactions on AI and Machine Learning, 2024 Publication	<1%

24% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups



31 AI-generated only **24%**

Likely AI-generated text from a large-language model.



0 AI-generated text that was AI-paraphrased **0%**

Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

