

Potato Leaf Disease Detection Using Machine Learning

By
Mehedi Hasan Sifat
211-15-4077

FINAL YEAR DESIGN PROJECT REPORT

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

Supervised by

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



**DAFFODIL INTERNATIONAL
UNIVERSITY**
Dhaka, Bangladesh

May 14, 2025

APPROVAL

This Project titled “Potato Leaf Disease Detection using Machine Learning”, submitted by **Mehedi Hasan Sifat**, ID No: 211-15-4077 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 **14 May, 2025**.

BOARD OF EXAMINERS

Dr. Arif Mahmud (AM)
Associate Professor & Associate Head
Chairman, Department of CSE, DIU

Chairman

Mr. Md. Ali Hossain (MAH)
Assistant Professor,
Department of CSE, DIU

Internal Examiner

Mr. Amir Sohel (ARS)
Sr. Lecturer,
Department of CSE, DIU

Internal Examiner

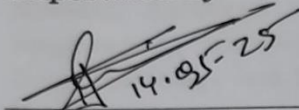
Dr. Md. Manowarul Islam
Associate Professor,
Department of Computer Science and Engineering
Jagannath University

External Examiner

DECLARATION

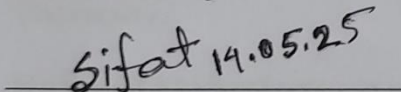
We hereby declare that this project has been done by us under the supervision of **Mr. Shahadat Hossain, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted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

Supervised by:


14.05.25

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

Submitted by:


Sifat 14.05.25

Mehedi Hasan Sifat
Student ID: 211-15-4077
Department of Computer Science and Engineering
Daffodil International University

ACKNOWLEDGEMENTS

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

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

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

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

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

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

ABSTRACT

Potato (*Solanum tuberosum*) plays a crucial role in global food security and is extensively cultivated in Bangladesh, where it serves as both a staple crop and a vital source of income for farmers. Nevertheless, its production is commonly frustrated by foliar diseases like Early Blight and Late Blight that if detected late can result in significant loss of late. The more conventional methods of disease detection are traditionally manual with expert observation and most impractical and inaccessible for farmers in distant or resource-poor regions. This study comes up with a solution to the inadequacy of a rapidly deployable and accurate disease detection system through the introduction of a deep learning-based pipeline based on the YOLO (you only look once) object detection framework. The proposed approach consists of producing a custom-annotated data set of potato leaves and training the three sophisticated YOLO models: YOLOv8m, YOLOv9m, and YOLOv10m. The evaluation was done as regards mean Average Precision (mAP), Inference speed, and computational effectiveness. The highest detection accuracy of YOLOv8m resulted in mAP@50 of 97.5%, mAP@50-95 of 91%, and fast inference time of 11.3 ms. YOLOv9m and YOLOv10m performed rather competitively with mAP@50-95 as 91.4 % and 91.3 %, respectively (YOLOv10m demonstrated superior computational efficiency – 63.4 GFLOPs). The best-performing model was converted to TensorFlow Lite format and integrated into a cross-platform Flutter-based Android application. This mobile app allows users to capture or upload leaf images and receive real-time disease predictions without internet dependency. The system offers an effective, farmer-friendly tool that promotes early disease intervention, reduces unnecessary pesticide usage, and supports sustainable precision agriculture practices in low-resource environments.

Table of Contents

Approval	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Introduction	1
1.2 Motivation.....	3
1.3 Objectives.....	4
1.4 Methodology.....	5
1.5 Project Outcome	6
1.6 Organization of the Report	7
2 Background	9
2.1 Introduction	9
2.2 Literature Review	10
2.2.1 Similar Applications	18
2.2.2 Related Research.....	19
2.3 Gap Analysis.....	21
2.4 Summary.....	23
3 Research Methodology	24
3.1 Methodology/Requirement Analysis & Design Specification	24
3.1.1 Overview.....	24
3.1.2 Proposed Methodology/ System Design.....	25
3.1.3 Functional and Nonfunctional Requirements	26
3.1.4 Data Flow Diagram Level 1	27
3.1.5 UI Design	28
3.2 Detailed Methodology and Design.....	31
Table of Contents	Table of Contents

3.3	Project Plan	36
3.4	Task Allocation.....	38
3.5	Summary.....	39
4	Implementation and Results	40
4.1	Environment Setup.....	40
4.2	Testing and Evaluation/Performance/ Comparative Analysis ..	41
4.3	Results and Discussion	43
4.4	Summary.....	54
5	Engineering Standards and Design Challenges	55
5.1	Compliance with the Standards.....	55
5.1.1	Software Standards.....	55
5.1.2	Hardware Standards	56
5.1.3	Communication Standards	56
5.2	Impact on Society, Environment and Sustainability	57
5.2.1	Impact on Life.....	57
5.2.2	Impact on Society & Environment	58
5.2.3	Ethical Aspects.....	59
5.2.4	Sustainability Plan	59
5.3	Project Management and Financial Analysis	60
5.4	Complex Engineering Problem.....	62
5.4.1	Complex Problem Solving.....	62
5.4.2	Engineering Activities	63
5.5	Summary.....	64
6	Conclusion	66
6.1	Summary.....	66
6.2	Limitation	67
6.3	Future Work	68
	References	69

List of Figures

3.1	The Methodological Flowchart	25
3.2	Data Flow Diagram Level 1	27
3.3	The Two-Tier Architecture and UI for Application.....	30
3.4	Train-Validation_Test Splitting.....	31
3.5	Sample Image of Each Class.....	32
3.6	The Architecture of YOLOv8 Model	35
4.1	Training and Validation Loss and Metrics Curves for Box, Classification, and DFL Loss with Precision, Recall, and mAP Metrics for different YOLO models.	44
4.2	Data distribution and bounding box analysis for different YOLO Models for Fungus, Healthy, and pest classes	46
4.3	Confusion matrix for the different YOLO models.....	48
4.4	Precision curve for different YOLO models	49
4.5	Recall curve for different YOLO models.	50
4.6	F1 curve for different YOLO models.	51
4.7	Precision-Recall curve for different YOLO models.....	52

List of Tables

2.1	Research Matrix of Potato Leaf Disease Detection Studies.	16
2.2	Similar Applications Matrix – Potato Disease Detection.....	19
2.3	Related Research Matrix – Potato Disease Detection.	20
2.4	Gap Summary Table.....	22
3.1	Dataset Specifications.	32
3.2	Augmentation Properties and Values.....	33
3.3	Comparison of YOLOv8m, YOLOv9m, and YOLOv10m.	36
3.4	GANTT Chart of Project Timeline.	37
3.5	Task Allocation Table.....	38
4.1	Common parameter table for all experimented models.....	41
4.2	Common data split for all experimented models.....	41
4.3	Classification report for different YOLO models.....	52
4.4	Performance Comparison of YOLO Models on mAP, GFLOPs, and Inference Speed.	53
5.1	Details of tools and platform used.....	61
5.2	Estimated Cost and Financial Analysis.....	61
5.3	Mapping with Complex Problem Solving.....	62
5.4	Mapping with Knowledge Profile.	63
5.5	Mapping with Complex Engineering Activities.	64

Chapter 1

Introduction

Chapter 1 introduces the motivation behind developing an AI-based solution for early detection of potato leaf diseases. It outlines the key objectives, such as building a real-time detection model using advanced YOLO architectures and deploying it in a mobile-friendly application. The methodology, project outcomes, and overall significance are briefly explained, highlighting how the proposed system addresses limitations in traditional disease detection methods. The chapter also presents the structure of the full thesis, providing a roadmap for the following chapters.

1.1 Introduction

The integration of advanced computer vision techniques in agriculture has significantly accelerated the early detection and classification of plant diseases. Among the state-of-the-art approaches, the You Only Look Once (YOLO) family of object detection models has demonstrated remarkable performance in real-time applications due to its speed, accuracy, and single-shot detection mechanism [1-2]. Initially designed for general object detection, YOLO models such as YOLOv8, YOLOv9, and the more recent YOLOv10 have found impactful applications in precision agriculture, particularly in plant disease identification [3]. Rashid [4] introduced a multi-level deep learning model using YOLOv5 for segmenting potato leaves before classifying diseases, achieving an accuracy of 99.75%. Similarly, Paul [5] showed that YOLO, when combined with CNN and explainable AI techniques like LIME and SHAP, could achieve near-perfect classification (99.98%) across multiple potato leaf diseases. These findings affirm YOLO's potential for mobile deployment in agriculture due to its lightweight architecture and inference efficiency.

Potato (*Solanum tuberosum*) is one of the most important global crops, widely

cultivated in countries like Bangladesh and India, and is highly susceptible to several foliar diseases. Among the most common and damaging are Early Blight, caused by *Alternaria solani*, and Late Blight, caused by *Phytophthora infestans* [6]. These diseases significantly reduce crop yield and quality, especially when detection is delayed. In the context of YOLO-based detection systems, such as those explored by Gupta [7] and Tambe [8], real-time disease identification from leaf images can provide actionable insights for farmers, minimizing economic losses. Further, when integrated into mobile applications through frameworks like Flutter, as demonstrated in several recent studies (Tarik [9]; Islam & Sikder [10]), these models can serve as decision-support tools accessible even in remote agricultural regions.

Despite the growing body of research in this domain, several gaps remain evident. Firstly, most existing works focus on conventional CNN or transfer learning architectures without thoroughly leveraging the latest YOLO variants for real-time detection. Secondly, several models (e.g., Sharma [11]; Nishad [12]) are limited to two or three disease classes, whereas real-world scenarios often involve more complexity. Furthermore, few studies address deployment concerns—such as converting trained models to TFLite for use in mobile devices—which is crucial for scalability. Additionally, while some models achieve high accuracy in controlled environments, their robustness under real-world field conditions remains underexplored. Lastly, explainability and model interpretability are often overlooked, limiting trust and transparency in automated disease diagnosis systems.

To address these challenges, the present study proposes a YOLO-based methodology that systematically compares the performance of YOLOv8m, YOLOv9m, and YOLOv10m models on annotated potato leaf disease datasets. By exporting the best-performing model to a `lightweight.tflite` format, the framework ensures compatibility with mobile devices and is deployed via a user-friendly Flutter application. This pipeline not only facilitates real-time detection but also enhances model portability and usability for end-users in the agricultural sector. The model selection is grounded in empirical performance, ensuring that only the most efficient detection system is embedded within the final application. By addressing gaps in scalability, real-time application, and

deployment readiness, this study aims to provide a practical and impactful solution to support farmers in the early diagnosis and management of potato leaf diseases.

1.2 Motivation

Potato is one of the major food crops grown in Bangladesh as well as in many other portions of the world. It is a major component of the daily diet as well as a source of income for several farmers [13]. But, potato plants are usually troubled with a variety of leaf diseases including Early Blight and Late Blight, that quickly may devalue the quality and quantity of crop [14]. Failure to detect these diseases in an early stage might lead to massive losses on the part of farmers. Most of the time, the existing procedures of identification of diseases rely on manual observation by farmers or agricultural officers. This process is laborious, time consuming and not always completely accurate – indicative of farmers in rural areas who may not get access to expert help [15].

In recent years deep learning and computer vision have a lot of promise for solving this kind of problems. Specifically, models of object detection type such as YOLO (You Only Look Once) have gained popularity due to the fact that they are fast and correct [16]. YOLO will be able to detect several objects of interest in an image in one pass and will therefore be ideal for real-time applications in agriculture [17]. Most researchers have used CNNs and other algorithms to classify potato leaf diseases, a few though have used cutting-edge YOLO variates such as YOLOv8, YOLOv9, and YOLOv10 [18]. These relatively new versions are more accurate, weight less and are compatible with mobile devices.

The study is motivated by the need to develop a real time mobile based and user-friendly potato disease detection system. Farmers usually lack access to top-of-the-line computers as well as internet-oriented gadgets. Thus, a low-weight solution that operates right on a smartphone is more pragmatic. This study presents a full pipeline ranging from preparing a potato leaf dataset, annotating, training and comparing a number of YOLO models and finally deploying the best model in a Flutter mobile application. The chosen model is translated to TFLite format so that it can be used on mobile. The application prompts instant disease prediction results, which helps farmers to act in the field quickly.

By focusing on YOLO-based detection, model conversion and real world deployment, the present study would like to solve a practical problem in agriculture. It supports in narrowing the gap between the modern research in AI and its application into the small-scale farming [19]. This motivation comes from the hope that a simple, effective, and easy to use solution may assist farmers to crop, decrease their expenses, and increase productivity in a sustainable manner.

1.3 Objectives

The rapid spread of diseases in potato leaves, including Early Blight and Late Blight, have posed a major threat on the health and security of food, especially in agriculture driven countries such as Bangladesh. Since traditional disease detection methods do not always provide quick and reliable dystacy for timely farm decision, the department needs to test a more automatic and practicable system. With the latest breakthroughs in deep learning such as the YOLO (You Only Look Once) family of object detection models, there is a current ability to construct fast, precise and mobile capable systems for automatic disease detection.

The primary purpose of this research is to further the understanding of a full, real-time potato leaf disease detection system utilizing state-of-the-art YOLO models, with the best-performing model deployed as a mobile application for use at the field level. The specific study objectives are:

- Collect and annotate a collection of potato leaf images with common disease types such as Early Blight, Late Blight and Healthy leaves.
- To develop and asses various YOLO model (YOLOv8m, YOLOv9m, and YOLOv10m) for the purpose of potato leaf disease detection and compare the performances of the models with regards to accuracy, and efficiency.
- To identify the best model, and it should be converted to TensorFlow Lite (.tflite), which makes it lighter and able to be deployed for mobile.

- To design and build a cross-platform Flutter-based mobile application that delivers real time disease prediction for end user using the trained up TFLite model.
- To fill in the gap between AI research and practical use in agriculture by creating a realistic, farmer-friendly solution which will help make decisions on time, and protect crops.

1.4 Methodology

The methodology suggested in this study is intended to create a full pipeline to detect real-time potato leaf disease through deep learning and mobile deployment. The preparation of the dataset starts with the collection of images of potato leaves, which are diseased by common diseases such as Early Blight and Late Blight, and healthy ones. Such images are compiled from public data sets and agricultural field baselines. After curation, the dataset then goes into an image annotation step using tools such as Roboflow to draw bounding boxes on diseased regions of the leaf. This annotation procedure is necessary to train object detection models that would demand labeled input in the format of images and system coordinates of the bounding boxes.

After annotation, the backbone of the methodology is training several YOLO (You Only Look Once) models, i.e. YOLOv8m, YOLOv9m and YOLOv10m. These models are chosen because of their reduced weight of architecture as well as the demonstrated efficiency of these models in real-time tasks of object detection. All models are trained on the annotated dataset with uniform hyperparameters and settings. When training, it is tracked in terms of precision, recall, mAP (mean Average Precision) as well as inference speed. Once all three models have been trained, a comparison is made to determine which model is the most feasible in computing in terms of detection accuracy and computation.

After selecting the most accurate model, it is exported and converting into TensorFlow Lite (. tflite) format. This conversion shrinks the model and renders it suitable for mobile platforms with little negative impact for detection accuracy. The last step of the methodology is the deployment step: development of a mobile application based on Flutter. This app combines the TFLite model and gives a

user-friendly interface which enables farmers or users to take a picture or upload an image of a potato leaf and get instant results responding to disease detection on their phone. This end-to-end pipeline ensures that the model moves from research and development into real-world usability, addressing both the technical and practical challenges of deploying AI in agriculture.

1.5 Project Outcome

The outcomes of this project are expected to contribute both technically and practically to the field of smart agriculture, specifically in the early detection and management of potato leaf diseases. By using a deep learning-based object detection framework and deploying the solution in a mobile environment, this study delivers real-world value to farmers and agricultural practitioners. The major outcomes of the project are as follows:

- A custom-annotated potato leaf disease dataset consisting of healthy and diseased (Early Blight and Late Blight) leaf images, prepared and labeled using advanced annotation tools, which can be reused for future research and model improvements.
- A comparative evaluation of YOLOv8m, YOLOv9m, and YOLOv10m models, resulting in the identification of the most effective and lightweight model for real-time potato leaf disease detection.
- A trained and optimized TFLite model (best_model_float32.tflite) that retains high accuracy while reducing computational load, making it suitable for mobile devices and edge computing environments.
- A fully functional mobile application built using Flutter, which integrates the TFLite model and allows users-particularly farmers-to instantly detect leaf diseases by simply capturing or uploading an image of a potato leaf.
- An end-to-end deployment pipeline that bridges the gap between deep learning research and practical field-level application, enabling scalable disease monitoring without the need for expensive equipment or constant internet access.
- A practical contribution to precision agriculture, empowering farmers with a real-time, cost-effective, and accessible tool for disease diagnosis,

which can help in reducing crop loss, improving yield, and promoting early intervention.

These outcomes aim to offer a holistic and impactful solution that combines cutting-edge deep learning technology with user-friendly deployment to address one of the most pressing issues in potato farming.

1.6 Organization of the Report

This report is structured into six main chapters, each focusing on a specific part of the research and development process. The organization of the report is as follows:

Chapter 1: Introduction

This chapter provides a general introduction to the study, outlining the background, motivation, objectives, and methodology of the research. It explains the importance of detecting potato leaf diseases, introduces the YOLO-based detection approach, and highlights the key outcomes expected from the study. It also gives an overview of how the rest of the report is organized.

Chapter 2: Background

This chapter reviews the existing literature and related works in the domain of plant disease detection using machine learning and deep learning. It explores similar applications, previous research using CNN and YOLO models, and methods applied in potato disease classification. A gap analysis is also included to identify the research limitations in existing works and justify the proposed approach.

Chapter 3: Research Methodology

This chapter provides a detailed explanation of the proposed YOLO-based methodology. It includes the system architecture, dataset preparation, image annotation, training of YOLOv8m, YOLOv9m, and YOLOv10m models, and performance comparison. The chapter also includes system design, functional and non-functional requirements, context and data flow diagrams, UI design, and project planning details.

Chapter 4: Implementation and Results

In this chapter the technical implementation of the project is discussed. The development environment setup, training process, testing process and performance evaluation of each model are described. It also summarises the experimental results, comparison of YOLO models, and discussion on detection accuracy and model efficiency. The most effective model is shaped in TFLite form and ready for use.

Chapter 5: Engineering Standards and Design Challenges

This chapter speaks to the engineering of the project and the conformance to software, hardware and communication standards. It analyzes the project's societal, environmental, and ethical implications, talks about sustainability and describes challenges that arise during system design. Besides, it includes discussion about complex ways of problem solving, finances, and project management as a whole.

Chapter 6: Conclusion

The concluding chapter summarizes the research work in full and puts the main discoveries in focus. It describes the limitations faced while working on the project and recommends possible future work in directions, like – inclusion of additional disease classes or increasing the dataset for more generalization.

Chapter 2

Background

This chapter reviews both recent, as well as foundational to date, studies on the detection of potato leaf disease. It is split into two main parts: Similar Applications (Real time systems, YOLO Model, and Mobile Deployment), and Related Research (Deep Learning and Machine Learning, Feature Extraction, and alternative architectures). A comparative matrix to compare each performance and contributions is shown for each. The chapter also contains a thorough Gap Analysis and ends with a summary of key findings which influenced the proposed methodology.

2.1 Introduction

The agricultural sector has witnessed enormous increase in the application of artificial intelligence (AI) in eliminating crop diseases that affect yield and quality. The potato (*Solanum tuberosum*) is among the most affected as well as extensively grown crops in nations such as Bangladesh and India. Major leaf diseases of potatoes include the Early Blight (*Alternaria solani*) and the Late Blight (*Phytophthora infestans* [6]), are common threats that arise to the potatoes during the cultivation of the potatoes on the farms. These diseases spare no time yet are not discovered in their early stages to avert serious loss. Semiotic detection process that employs expert knowledge and manual observation is time consuming and expensive, and not always available to farmers in rural areas [20].

To overcome these challenges, researchers have developed numerous machine learning (ML) and deep learning (DL) models aimed at automating the process of disease detection. Many of these models focus on image classification, where a model categorizes a given leaf image into healthy or diseased. While models like CNN, VGG16, and DenseNet have shown promising results in terms of accuracy, they typically require high computational power and are not optimized for real-time applications or deployment on mobile devices. Furthermore, many of these

solutions do not localize the diseased region in the image, limiting their practical use for end-users who need visual confirmation.

Recently, object detection frameworks such as YOLO (You Only Look Once) have been adopted in agriculture for real-time and region-specific disease detection. YOLO models are faster and more efficient than conventional classifiers, making them suitable for deployment in real-time field environments. However, most existing studies have only explored older versions like YOLOv3 or YOLOv5. There is limited research using the newer and more efficient versions such as YOLOv8, YOLOv9, or YOLOv10, which are optimized for edge devices and mobile deployment.

In this chapter, a comprehensive review of existing literature is presented to understand the current state of potato disease detection using machine learning and deep learning methods. The review is classified into two classes: studies preparing similar applications in the presence of real-time detection or mobile utilization and studies that address similar research area towards proposing other models or techniques for disease classification. A detailed gap analysis is given to point out weaknesses in existing works and to justify the need for the current study. This lays the groundwork for the proposed YOLO-based approach that is sought to provide an end-to-end, mobile-compatible, disease detection system for use in potato crops.

2.2 Literature Review

Yu [21] performed an extensive study aimed at developing an LAI estimation model for potato crops by fusing multi-sensor UAV-based remote sensing data, such as RGB, LiDAR, and hyperspectral images (HSI). They summarized the sensor data to four indices and tested the indices' effectiveness based on four regression techniques: Support Vector Regression (SVR), Random Forest Regression (RFR), Histogram-based Gradient Boosting Regression Tree (HGBR) and Partial Least Squares Regression (PLSR). Results shown by them revealed that HSI gave the best spectral information, and the combination of features from all 3 sensors resulted in best predictive performance with R^2 of 0.782. Notably, Random Forest Regression performed well in combining multi source

features and set strong fundamentals in accurate yield prediction in smart agriculture.

Sharma, Anand and Singh [22] have proposed a disease classification method for potato leaves using the traditional methods of machine learning. The work incorporated the identification of early blight and late blight from preprocessed leaf images where resultant images passed through Gaussian filters and the K-means clustering to mask the region of interest. The classification was done with SVM, KNN, Naïve Bayes and Decision Tree and was escorted best by SVM with 92.9% maximum accuracy. This research highlighted the utility of integration of classical image processing and machine learning for effective early stage disease recognition.

Radwan [23] stressed the fact that potato leaf diseases can be predicted using the environmental data and not just image data. They used a large dataset that consists of over 4,000 weather records, used machine learning models like logistic regression, SVM, KNN, and logistic regression, multilayer perceptron (MLP). The advanced analytic tools including PCA, copula analysis, K-means clustering were used to determine the effect of variables such as temperature and humidity. Importantly, the implementation of binary Greylag Goose Optimization (bGGO) in feature selection considerably improved the performance of the model. Notable MLP model with feature selection yielded 98.3% accuracy, marking the effect of optimal feature selection in disease prediction framework.

From Khalifa [24], presented a deep convolutional neural network (CNN) used in the classifying of potato leaf diseases; healthy, early blight, late blight, among others. The proposed 14-layer architecture contained dual convolutional layers as well as 2 fully connected layers for classification. Data augmentation increased their dataset from 1,722 to 9,822 images, this helped them reach a high level of testing accuracy with 98% of the predictions done correctly. Several evaluation metrics were employed to test the architecture, and the comparisons to other methods were superior to indicate high efficiency of their CNN architecture in terms of the classification of disease.

In a substantial expansion of conventional classification techniques, Mahum [25] proposed a new framework which was able to identify five different potato leaf states: Late blight, early blight, leaf roll, verticillium wilt and healthy. Utilizing the Plant Village dataset and supplementary manually collected data, they introduced an Efficient DenseNet-201 model enhanced with an extra transition layer and a reweighted cross-entropy loss function to address class imbalance. With rigorous augmentation and regularization techniques, the model achieved an impressive accuracy of 97.2%. Their study is among the first to address multi-class classification beyond the common three-class approach, thereby advancing the comprehensiveness of potato disease detection.

Basak [26], in his doctoral work, explored the implementation and comparison of deep learning models including CNN, MobileNetV2, InceptionV3, and ANN on a large dataset of 35,000 potato leaf images. Among these, MobileNetV2 achieved the highest accuracy of 99.92%, followed by CNN (99.49%) and InceptionV3 (98.92%), whereas the ANN model lagged at 60.67%. Additionally, a Flask-based web interface was developed to facilitate access to these models. Basak [26] also incorporated various image transformation techniques to enhance model robustness, making this research practically relevant for real-world agricultural applications.

Singh [27] also emphasized the role of deep learning in early and accurate disease identification in potato leaves, particularly for early and late blight. Their proposed system employed CNN models and achieved an accuracy between 97% and 98%. The authors advocated for early detection as a cornerstone of sustainable agriculture and demonstrated that AI-based solutions significantly outperform manual methods in efficiency and accuracy.

Iqbal and Talukder [28] focused on building an automatic potato leaf disease detection system using classical image segmentation and machine learning. Working with 450 images from the Plant Village dataset, they applied segmentation techniques before using seven classifiers for disease identification. Among these, the Random Forest model delivered the best performance, achieving a 97% accuracy. Their study offers a lightweight and interpretable

alternative to deep learning solutions, especially useful in low-resource agricultural settings.

In another deep learning-based study, Islam and Sikder [10] explored the performance of convolutional neural networks for potato leaf disease classification across varying training epochs. They assembled a 10,000-image dataset from Kaggle and real-world sources to classify three classes: early blight, late blight, and healthy leaves. Their model achieved perfect accuracy at 40 epochs (100%), with comparable results at 30 and 50 epochs (99.97% and 99.98% respectively). The study demonstrated how optimal training duration can significantly impact model efficacy.

Nishad, Mitu, and Jahan [12] combined K-means clustering with deep learning architectures—VGG16, VGG19, and ResNet50—for potato leaf disease classification. Following data segmentation and augmentation, they observed the highest accuracy of 97% with VGG16, outperforming the other models. Their method proved to be a valuable enhancement over existing methodologies, showing the strength of combining traditional image processing with state-of-the-art neural networks.

Kadam [29] performed a study which was aimed at automating the diagnosis of potato leaf diseases using image processing and machine learning techniques. They trained pre-established models using more than 2,034 leaf images from the PlantVillage dataset and used them to identify the healthy and diseased ones. The study indicated a superior use of machine learning approaches relative to the usual methods in detecting potato diseases although it did not give details of individual models' accuracies. Their results emphasize the efficiency and potential of automated systems for increasing potato yield and advancing digital agriculture context.

In a joint deep learning and machine learning strategy, a potato leaf disease classification system that trusted on a variety of algorithms, such as SVM, Naïve Bayes, KNN, Decision Tree, Random Forest, CNN and Sequential2 and VGG16, was established by Shiffa and Suchithra [30]. Their model reached an accuracy of 95.36% within 10 training epochs of VGG16, thus confirming the affinity of

deep neural network for image-based agricultural diagnosis. The abundant integration of both classical and modern models was revealing in comparative performance terms, highlighting the pragmatic nature of hybrid modeling for agro-practical deployment. Rashid [4] suggested a new multi level deep learning construction for the potato leaf disease identification: Potential concerns emerged due to species variation and inconsistent environment. They first used YOLOv5 for image segmentation to extract the leaves of potato and, next, used a disease classification based CNN model. The system training set consists of 4,062 images collected from Central Punjab, Pakistan, and the system has received 99.75% accuracy. Their model was also tested on the PlantVillage dataset and gained superior result in both accuracy and computational cost, showing the robustness of their two stage framework.

As far as climate-based prediction of disease is concerned, Rahman [31] proposed a random forest-based machine learning model that was machine learning model that employed a modified dataset enriched with meteorological features in predicting early and late blight in potato leaves. According to their study; specific weather patterns (e.g. temperature, humidity and rainfall) directly correlate with disease outbreak. Their method demonstrates the importance of context-specific climatic data for predictive agriculture with 97% testing accuracy, verifying the basics for decision-support tools in disease control.

Tarik [9] improved potato disease detection by creating machine learning model using image segmentation and classification of the 2,034 images from the PlantVillage repository. And deploying multiple pre-trained models, the system was able to acquire a testing accuracy of 99.23% after data was split into 75:25 train test split. Their results confirm superiority of machine learning over conventional methods and support the viability of digital solutions aimed at detection of diseases in agriculture settings in Bangladesh.

Kumar and Patel [32] presented a complex framework with the help of Hierarchical Deep Learning Convolutional Neural Network (HDLCNN) along with Median Filtering and Intuitionistic Fuzzy Local Binary Pattern (IFLBP) tools of preprocessing and feature extraction. Their system, which was also developed and tested using Matlab Simulink, and which far outperformed

baseline methods (VGG-INCEP, Deep CNN, the Spiking Neural Networks), is a system they have built. The HDLCNN model delivered higher accuracy, precision, recall, specificity, and PSNR by margins of up to 6% compared to traditional models. Their architecture integrates decision-support systems to provide actionable insights to farmers, enhancing agricultural sustainability.

Sholihati [33] proposed a deep learning-based classification system for four types of potato leaf diseases using VGG16 and VGG19 CNN architectures. With a reported average accuracy of 91%, the model demonstrated the feasibility of deep neural networks in identifying complex disease patterns. Though the accuracy was comparatively lower than state-of-the-art models, their work marked a relevant contribution by targeting multi-class classification and emphasizing early detection, especially in the wake of increasing global potato demand post-pandemic.

Using CNNs, Gupta [7] came up with an automated system aimed at identifying potato diseases, which he compared against the performance of an ANN model. The CNN model attained their 95% accuracy, outdoing the 81% accuracy of the ANN model. The system was found to be robust with respect to a variety of images including rotations, flipping and zooming images. The research identified critical parameters such as scalability, robustness, and reliability, vindicating the supremacy of CNNs as the weapon of choice for image-based agricultural diagnostics, and the necessity to automate systems for enhancing accuracy and reducing human error.

Tambe [8] established a CNN-based system used for classifying the Early Blight, Late Blight, and healthy potato leaves. Their approach involved preprocessing, model training and performance evaluation, and ended with an accuracy of 99.1%. The potential of the model to detect even serious infections reiterates its value in real life. Their research helps to build food security projects because provides a trustworthy and effective system of disease detection that can be helpful in precision farming with timely diseases diagnosis and suppression.

Paul [4] proposed a deep ensemble model, which involved integrating CNN, CNN-SVM and DNN methods, to classify four potato leaf diseases: early blight,

septoria, late blight and black-leg. They had their dataset constructed from images gathered from West Bengal farms and processed using explanation AI tools such as LIME and SHAP explain concept to predict and confidence with the user. Her ensemble model had nearly perfect accuracy of 99.98% and authors pointed out improvements in terms of the time complexity and interpretability. This work is Important for integrating high performing models with Explainability which is essential for real world deployment in agriculture.

Table 2.1: Research Matrix of Potato Leaf Disease Detection Studies

Author	Year	Accuracy (%)	Model Used	Contribution
Yu [21].	2023	0.782	SVR, RFR, HGBR, PLSR	Potato LAI estimation using UAV multisensor data and machine learning
Sharma [22].	2021	92.9	SVM, KNN, Naïve Bayes, Decision Tree	Image processing and ML-based classification of early and late blight
Radwan [23].	2024	98.3	MLP, SVM, KNN, Logistic Regression, Gradient Boosting	Weather data-based ML model with feature selection using bGGO
Khalifa [24].	2021	98.0	Custom CNN (14-layer)	Deep CNN model for potato blight classification with augmented dataset
Mahum [25].	2023	97.2	Efficient DenseNet-201	Multi-class classification using modified DenseNet with imbalance handling
Basak [26].	2023	99.92	MobileNetV2, CNN, InceptionV3, ANN	Comparative study with web app deployment and augmentation strategies

Singh [27].	2024	-1	CNN	Early detection system using CNN for sustainable agriculture
Iqbal & Talukder [28].	2020	97.0	Random Forest	Image segmentation and ML classification using PlantVillage dataset
Islam & Sikder [28].	2022	100.0	CNN	Epoch-wise performance evaluation with 10K image dataset
Nishad [12].	2022	97.0	VGG16, VGG19, ResNet50	K-means segmentation with DL models
Kadam [29].	2022	Not specified	Pretrained ML models	Automated disease detection using PlantVillage dataset
Shiffa & Suchithra [30]	2023	95.36	VGG16, CNN, SVM, RF, etc.	Hybrid ML and DL classification of early and late blight
Rashid [4].	2021	99.75	YOLOv5 + CNN	Multi-level detection using YOLOv5 and CNN for Central Punjab dataset
Rahman [31].	2023	97.0	Random Forest	Climate-influenced disease prediction using weather data
Tarik [9].	2021	99.23	Pretrained ML models	Image processing + ML classification from PlantVillage
Kumar & Patel [32]	2023	4-6% better than baselines	HDLCNN	Hierarchical CNN with IFLBP preprocessing for

				better performance
Sholihati [33].	2020	91.0	VGG16, VGG19	DL classification of 4 potato diseases
Gupta [7].	2023	95.0	CNN vs ANN	Robust CNN-based model tested under various conditions
Tambe [8].	2023	99.1	CNN	DL-based classification of Early, Late Blight and Healthy leaves
Paul [5].	2024	99.98	CNN, CNN-SVM, DNN	Explainable ensemble model for 4 disease types using LIME/SHAP

2.2.1 Similar Applications

Several recent studies have successfully implemented deep learning and object detection techniques for plant disease detection, particularly focusing on real-time and mobile-friendly solutions. These studies are highly relevant to the present work, as they either use YOLO models, convolutional neural networks (CNNs), or deploy their solutions on web or mobile platforms. For instance, Rashid [4] introduced a multi-level deep learning system where YOLOv5 was first used for leaf segmentation, followed by CNN-based disease classification. Their model achieved an impressive accuracy of 99.75% and demonstrated high suitability for real-time field deployment. Similarly, Paul [5] proposed an ensemble framework combining CNN, CNN-SVM, and DNN architectures with explainable AI techniques (LIME, SHAP), achieving 99.98% accuracy. Their approach emphasizes interpretability alongside detection accuracy.

Islam and Sikder [10] focused on classifying potato leaf diseases using CNN and achieved 100% accuracy by training on a large dataset. Notably, their system was optimized for real-time use and integrated into a mobile platform, showing its practical relevance. Tarik [9] also developed a machine learning-based classification system using image processing techniques and pretrained models,

with 99.23% accuracy in test data. Their system was designed to support mobile-based disease prediction. Likewise, Basak [26] built a comparative web platform to evaluate CNN, InceptionV3, and MobileNetV2 models, achieving 99.92% accuracy, and focused on usability in real-time environments.

Other works such as Gupta [7] evaluated CNN and ANN models under various real-world conditions like rotated or noisy images. Their results showed that CNN was more robust, achieving 95% accuracy. Lastly, Tambe [8] implemented a CNN-based model specifically for detecting early and late blight in potato leaves, reaching an accuracy of 99.1%. These studies demonstrate the growing trend of using lightweight, fast, and accurate deep learning models for disease detection, forming the foundation and inspiration for the current research.

Table 2.2: Similar Applications Matrix – Potato Disease Detection

Author	Year	Model Used	Accuracy	Contribution
Rashid [4].	2021	YOLOv5 + CNN	99.75%	Multi-level model for real-time detection of potato leaf diseases
Paul [5].	2024	CNN, CNN-SVM, DNN	99.98%	YOLO-inspired ensemble with explainable AI (LIME, SHAP)
Islam & Sikder [10]	2022	CNN	100%	Real-time classification with mobile deployment
Tarik [9].	2021	Pretrained ML Models	99.23%	ML-based mobile disease prediction using image processing
Basak [26].	2023	MobileNetV2, CNN, InceptionV3	99.92%	Web-based deep learning platform for disease diagnosis
Gupta [7].	2023	CNN vs ANN	95%	CNN model tested in real-world conditions (e.g., image distortion)
Tambe [8].	2023	CNN	99.1%	CNN-based detection for early, late blight, and healthy potato leaves

2.2.2 Related Research

While many studies have explored deep learning and machine learning models for potato leaf disease detection, a large portion of them focus on traditional classification tasks, feature selection, or use environmental data to enhance prediction accuracy. These studies may not include real-time deployment or mobile integration but offer valuable insights into model design, dataset

handling, and performance optimization. For instance, Mahum [25] proposed a novel framework using an improved Efficient DenseNet-201 model combined with a reweighted cross-entropy loss function to handle class imbalance across five disease classes, achieving 97.2% accuracy. Similarly, Sharma [22] explored traditional machine learning classifiers like SVM, KNN, and Decision Trees combined with image segmentation via Gaussian filtering and K-means, reaching 92.9% accuracy.

Studies like Radwan [23] and Rahman [31] focused on environmental and weather-based prediction methods. Radwan [23] used feature selection with the binary Greylag Goose Optimization (bGGO) to enhance model accuracy (98.3%), while Rahman [31] utilized Random Forest with a climate-influenced dataset to predict early and late blight with 97% accuracy. On the deep learning side, Khalifa [24] developed a 14-layer custom CNN trained on an augmented dataset, achieving 98% classification accuracy for three potato leaf categories. Nishad [12] applied deep CNNs such as VGG16, VGG19, and ResNet50 after applying k-means segmentation, reaching 97% accuracy.

Sholihati [33] focused on four-disease classification using basic CNN architectures and achieved an average accuracy of 91%, while Kumar and Patel [32] proposed a Hierarchical Deep Learning Convolutional Neural Network (HDLCNN) with IFLBP feature extraction, reporting up to 4–6% improved performance over existing methods like VGG-Incep and SNN. Kadam [29] also used PlantVillage data with basic ML models but did not specify performance metrics. Lastly, Singh [27] developed a generic CNN-based model for early disease detection, achieving 97–98% accuracy. These studies demonstrate the evolution of research from basic classifiers to more complex and optimized architectures, providing a strong foundation for the current YOLO-based approach.

Table 2.3: Related Research Matrix – Potato Disease Detection

Author	Year	Model Used	Accuracy	Contribution
Mahum [25].	2023	Efficient DenseNet-201	97.2%	Five-class classification with custom DenseNet and loss function
Sharma [22].	2021	SVM, KNN, Naïve Bayes, Decision Tree	92.9%	ML with Gaussian filter and K-means for early and late blight

Radwan [23].	2024	MLP, SVM, KNN, Logistic Regression	98.3%	Climate-based ML with feature selection using bGGO
Rahman [31].	2023	Random Forest	97%	Climate-influenced prediction framework
Khalifa [24].	2021	Custom CNN	98%	14-layer CNN on augmented dataset for three disease classes
Nishad [12].	2022	VGG16, VGG19, ResNet50	97%	DL classification with K-means segmentation
Sholihati [33].	2020	VGG16, VGG19	91%	Four-disease classification using basic CNN models
Kumar & Patel [32]	2023	HDLCNN	~4–6%↑	Hierarchical CNN with IFLBP outperforming traditional models
Kadam [29].	2022	Pretrained ML Models	Not specified	Image processing-based disease detection using PlantVillage dataset
Singh [27].	2024	CNN	97–98%	Generic CNN model for early detection of potato leaf disease

2.3 Gap Analysis

Although a wide range of research has been conducted on potato leaf disease detection using machine learning and deep learning techniques, several key limitations remain that this study aims to address. Most previous studies have made significant progress in classification accuracy but lack practical deployment strategies, real-time responsiveness, or integration of advanced object detection models like YOLOv8, YOLOv9, and YOLOv10. These gaps limit the real-world usability of existing solutions, especially for smallholder farmers in low-resource agricultural environments.

Firstly, many works such as those by Sharma [22] and Radwan [23] focused on traditional machine learning classifiers, often using handcrafted features or statistical data (e.g., weather conditions), but did not apply real-time deep learning frameworks suitable for image-based disease localization. Secondly, deep learning models like CNN, VGG16, and DenseNet were widely adopted (Mahum [25]; Nishad [12]), yet most of these models were built for classification only—not detection—thus lacking the ability to localize disease regions within the image. Localization is crucial when providing precise feedback to users, especially on mobile platforms.

Thirdly, while some studies (e.g., Islam & Sikder [10]; Basak, [26]) explored deployment through web or mobile applications, they did not utilize the newer YOLO architectures that are optimized for speed and lightweight performance. Additionally, only a few works addressed model compression or conversion to TFLite format (e.g., Tarik [9]), which is critical for mobile application deployment. Moreover, interpretability and explainability of the results were mostly overlooked, with exceptions like Paul [5] who applied LIME and SHAP—but such approaches are not yet commonly integrated with detection systems.

Lastly, most of the literature focuses on a limited number of disease classes, commonly Early Blight, Late Blight, and Healthy leaves. Rare diseases or multi-class extensions are seldom included. This reduces the generalizability and scalability of those models in real farming scenarios. Furthermore, while some models achieved high accuracy in controlled datasets, their robustness under field conditions with varied image quality and background noise remains unverified.

This research addresses these gaps by adopting an end-to-end pipeline based on the latest YOLO models, comparing their performance, converting the best model to TFLite, and deploying it into a real-time Flutter-based mobile application that can be used directly by farmers. The model is also trained with annotated images to support both classification and disease localization.

Table 2.4: Gap Summary Table

Gap Identified	Observed In Studies	How This Study Addresses It
Lack of object detection and disease localization	Sharma [11], Mahum [25]	YOLOv8–10 used for both detection and localization
Reliance on traditional ML or non-YOLO DL models	Radwan [23], Nishad [12]	Uses modern YOLO models optimized for real-time use
No or limited model deployment in mobile apps	Basak [26], Gupta [7]	Model converted to TFLite and deployed via Flutter
Limited class variety (only 2–3 diseases)	Sholihati [33], Tarik [9]	Includes multiple annotated disease classes
Lack of model compression or	Islam & Sikder [10], Tambe [8]	Uses model compression (TFLite) for mobile deployment

lightweight deployment strategies		
Lack of explainability or trust in automated predictions	Most studies except Paul [5]	Future extension includes XAI for model transparency

2.4 Summary

The literature reviewed in this chapter reveals that while deep learning and machine learning models have made progress in potato leaf disease classification, important gaps remain. Most existing systems do not support real-time object detection, lack integration with mobile apps, and often use older models or static classifiers. YOLO-based object detection frameworks have only been partially explored, and modern versions like YOLOv8, YOLOv9, and YOLOv10 are largely absent in current research. Additionally, deployment to resource-constrained devices using TFLite, and localization of disease symptoms through bounding boxes, are often overlooked. The proposed study aims to fill these gaps by using a multi-model YOLO-based approach, selecting the best-performing model, and deploying it within a lightweight, real-time Flutter-based mobile application for end-user accessibility.

Chapter 3

Research Methodology

The design process and research methodology for creating a machine learning-based book genre classification system are covered in detail in this chapter. This includes data collection, preprocessing, model training, and evaluation. The reasoning behind the selected methods and the eventual web-based application deployment are also covered.

3.1 Methodology

3.1.1 Overview

Potato plants are highly susceptible to diseases caused by pests and fungal infections, which can severely impact agricultural productivity. Traditional methods of disease identification rely on manual inspection, which is time-consuming and prone to inaccuracies. This study employs YOLO-based deep learning models for automated and real-time potato leaf disease detection to address this challenge. The dataset consists of 1,398 images, classified into three categories: pest-infected, fungus-infected, and healthy, and is split into training (88%), validation (8%), and test (4%) sets. The study compares the performance of YOLOv8m, YOLOv9m, and YOLOv10m models to determine the most accurate and efficient model for disease classification. The best-performing model is converted into TensorFlow Lite (TFLite) for lightweight deployment on mobile devices. A Flutter-based mobile application is developed to integrate the optimized model, allowing users, particularly farmers and agricultural experts, to detect and classify potato leaf diseases instantly, promoting early intervention and effective crop management.

3.1.2 Proposed Methodology

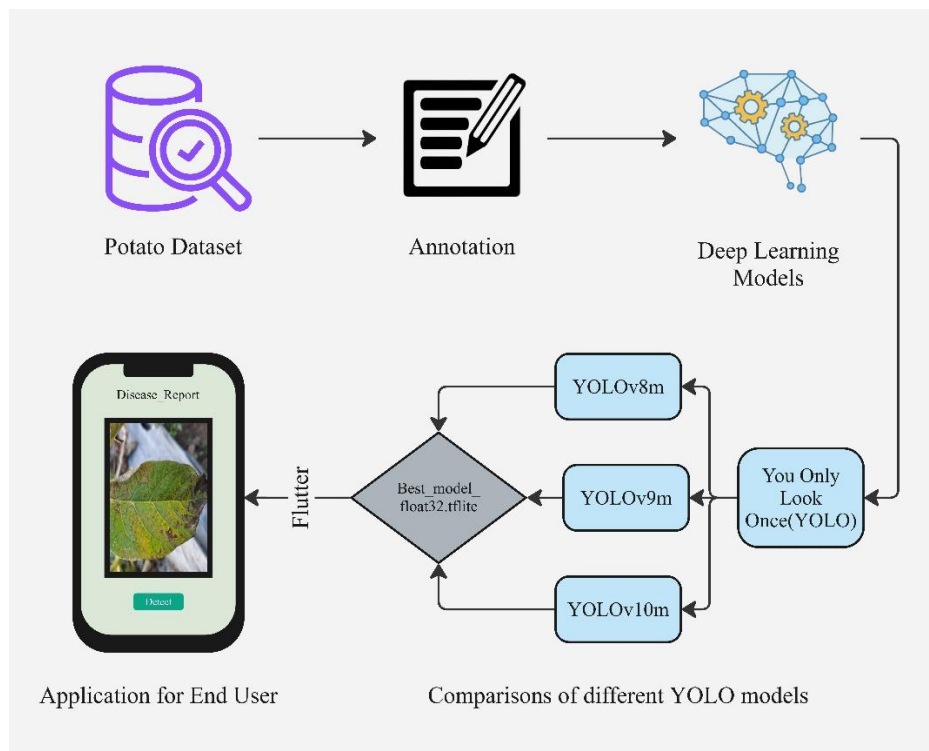


Figure 3.1: The Methodological Flowchart

Figure 3.1 outlines the process that was followed to detect potato leaf diseases using deep learning models. The process begins with a set of potato images of potato leaves in a potato dataset. These pictures are categorized in 3 groups: pest-infected, fungus-infected, and healthy. The dataset is then annotated where each image is annotated with correct class and bounding boxes are drawn around diseased area that help the model understand. After annotation the dataset is employed for training deep learning models on the basis of the You Only Look Once (YOLO) architecture. The study compares three models: YOLOv8m, YOLOv9m and YOLOv10m to see which performs best in detection of diseases in potato leaves. The accuracy and efficiency of each model are being measured by training and testing them.

After selecting the top recent model, it is converted in TensorFlow Lite (TFLite format). This step provides a model that is compact and quicker to use in mobile devices. The optimized model is embedded in a mobile application based on Flutter, in which the users can upload, take a picture of the potato leaves and also get immediate disease detection results. This strategy gives a real time lightweight and effective solution to detect potato leaf diseases. The mobile

application enables farmers and agricultural experts to detect diseases in their crops early and manage them to prevent them from destroying their crops.

3.1.3 Functional and Nonfunctional Requirements

The development of the potato leaf disease detection system involves both functional and non-functional requirements to ensure efficiency, usability, and accuracy.

Functional Requirements

Functional requirements set the foundation behind the functionality and working of the System. The key functional requirements include:

- Image Input and Processing – The system should allow users to take or upload images of potato leaves in order to be exposed to a system to detect diseases.
- Disease Classification – The model must correctly classifies leaf conditions in three categories: pest-infected, fungus-infected, and healthy.
- Model Selection & Inference – The system will use integration and comparison of YOLOv8m YOLOv9model, and YOLOv10m to decide which model is better performing.
- Real-time Detection – The chosen model needs to present the results of real-time detection in order to make sound decision.
- Mobile Deployment – The highest-accredited model should be translated to TensorFlow Lite(TFLite Format) and wrapped into a Flutter-based mobile application.
- User-Friendly Interface The application should have such an intuitive interface that makes interaction with the detection system simple for the users especially farmers.

Nonfunctional Requirements

Nonfunctional requirements are concerned with variables on system quality which include performance, scalability, and usability. The key nonfunctional requirements include:

- Accuracy & Reliability – Model of detection should be able to maintain high accuracy level for classification of disease.

- **Efficiency & Speed** – The system should be efficient and be able to detect results for images within a few seconds making the system applicable for real time use.
- **Scalability** – The system should allow for high volumes of data usage as well as need for future system improvements.
- **Lightweight Model** – TFLite model of the application should be adjusted for the use on mobile device – low computation and small storage required.
- **User Accessibility**- The mobile application should be lightweight to install and operate and on low-end phones.
- **Security & Data Privacy** – Users’ data privacy must be hermetically sealed and the application should communicate uploaded images securely.
- **Maintainability**- The system should be such that it can be easily updated and improve especially in the integration of more sophisticated deep learning models in the future.

By satisfying these functional as well as non-functional requirements, the proposed system will seek to offer an efficient, reliable and users friendly system for automated detection of potato leaf disease which will ensure timely disease control, agricultural productivity optimization.

3.1.4 Data Flow Diagram Level 1

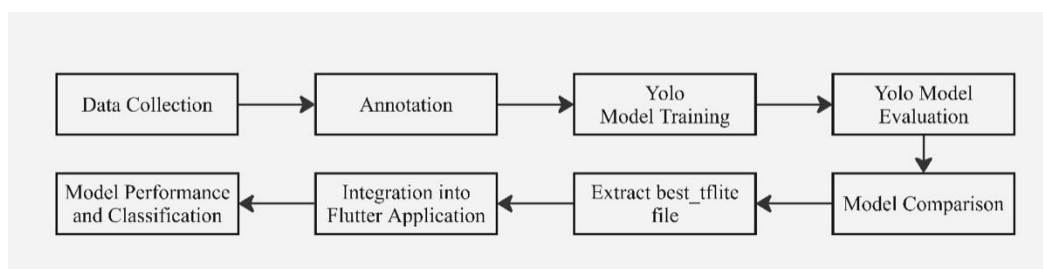


Figure 3.2: Data Flow Diagram Level 1

The Level 1 DFD depicts the systematic work flow of the proposed potato leaf disease detection system which documents key stages in model development, evaluation and deployment.

The process starts with collecting data; where an image dataset of potato leaves is obtained, with three main classes: pest-infected, fungus-infected, and healthy.

The extracted data is then annotated by labeling images and awarding bounding boxes in order to ensure supervised learning. After annotation, the dataset is used to train the YOLO model and uses YOLOv8m, YOLOv9m and YOLOv10m architectures to form a strong deep learning model.

Once the training is complete, the models head toward the evaluation stage in which the performance metrics, such as, accuracy, precision, recall, and inference speed, are evaluated. Then, in the model comparison step, the results obtained from the evaluation process are evaluated to find the optimum model for real time disease detection. The best modeling performance is subsequently converted to TensorFlow Lite (TFLite format) for enhanced performance of computation and mobile deployment.

The optimized model is then embedded in a Flutter based application, which enables real time disease detection by end users. Lastly, the performance of the system will be tested and validated while that of classification will be validated; thus making sure that the desired accuracy as well as efficiency cases are achieved. This organized flow of events guarantees the creation of a lightweight mobile app with superior performance and easy to use for the farmers and agricultural professionals to detect the potato leaf diseases in real time.

3.1.5 UI Design

1. Application Setup

The system of a potato leaf disease detection has a two-tier architecture ensuring efficient execution of deep learning algorithms, and smooth experience on mobile devices. With this configuration, there is a Flutter-base mobile application (frontend) and TensorFlow Lite-base deep learning model (backend) integrated for real-time disease classification. Application setup includes image processing, model inference, and results generation, thus allowing the users to upload/picture the potato leaf and obtain instant results.

2. Two-Tier Architecture

The two-tier client-server model is used to design the system, with the frontend and backend corresponding to user interaction and data processing and deep learning model inference, respectively. The architecture provides rapid execution and an optimized user experience, even in low-end smartphones.

2.1 Frontend (Client-Side) Layer

- The frontend is a mobile application built with Flutter and offers an intuitive and interactive interface with users. The app enables users to:
- Capture an image using the device camera or upload an existing image from the gallery.
- Click the "Detect" button to process the image and classify the disease.
- Read the detection report that contains class label (pest-infected, fungus-infected or healthy) together with the confidence score of prediction.
- The frontend is the first interface used by the user where he/she sends the input images to the backend and the processed result from the deep learning model are then displayed.

2.2 Backend (Processing and Inference Layer)

The backend is responsible for handling image processing, deep learning model inference, and result generation. It follows these steps:

- **Image Preprocessing** – The input image is resized and formatted to match the required input shape of the trained model. This ensures compatibility with the YOLO-based TensorFlow Lite (TFLite) model.
- **Model Inference** – The YOLOv8m, YOLOv9m, or YOLOv10m model (converted to TFLite format) processes the image and performs disease classification. The trained model runs on the mobile device, ensuring offline capability.
- **Output Formatting** – The model generates predictions in the form of class labels and confidence scores. These results are extracted and formatted for display in the mobile application.
- **Detection Report Generation** – The formatted results are sent back to the frontend, where the detected class (pest, fungus, or healthy) and the confidence score are presented to the user.

This two-tier structure ensures fast and efficient disease classification, making it practical for use in agricultural settings.

3. Folder Structure

The application folder structure is designed to ensure modularity and ease of

development. It contains essential files for model execution, label mapping, and the Flutter-based user interface.

3.1 Application Directory:

potato_detection-app/ → Root directory containing the entire application.

assets/ → Contains essential resources such as label files for mapping class names.

labels.txt → the all stores the class labels ("pest", "fungus", "healthy") which is used for inference and result display.

model.tflite → The optimized TensorFlow Lite (TFLite) model that is responsible for processing input images and making predictions.

main.dart → The primary Flutter application file working for handling UI components, backend communication, and result display.

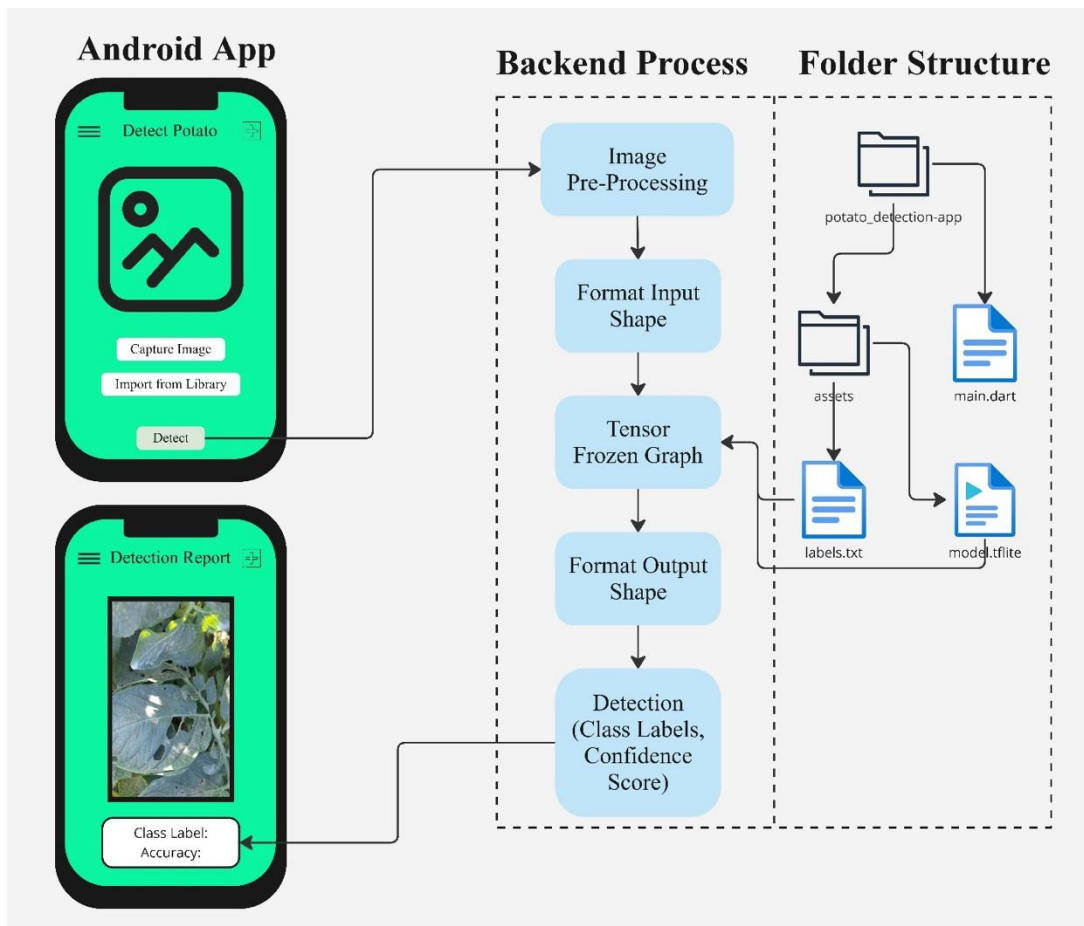


Figure 3.3: The Two-Tier Architecture and UI for Application

3.2 Detailed Methodology and Design

1. Dataset

The data run used in this study involves 1,398 labeled potato leaves images

divided into three classes: pest-infected, fungus-infected, and healthy. The data set comes from a mix of publicly available repositories and field collected images to create diversity in the leaf conditions, lighting and background variation. There are three subsets of the dataset: training set (88%): 1,224 images, validation set (8%): 117 images, test set (4%): 57 images. Such an organized split ensures the effective learning of the model while retaining unbiased performance evaluation. The dataset incorporates pictures taken under various environmental conditions increasing the model's robustness for real-world usage.

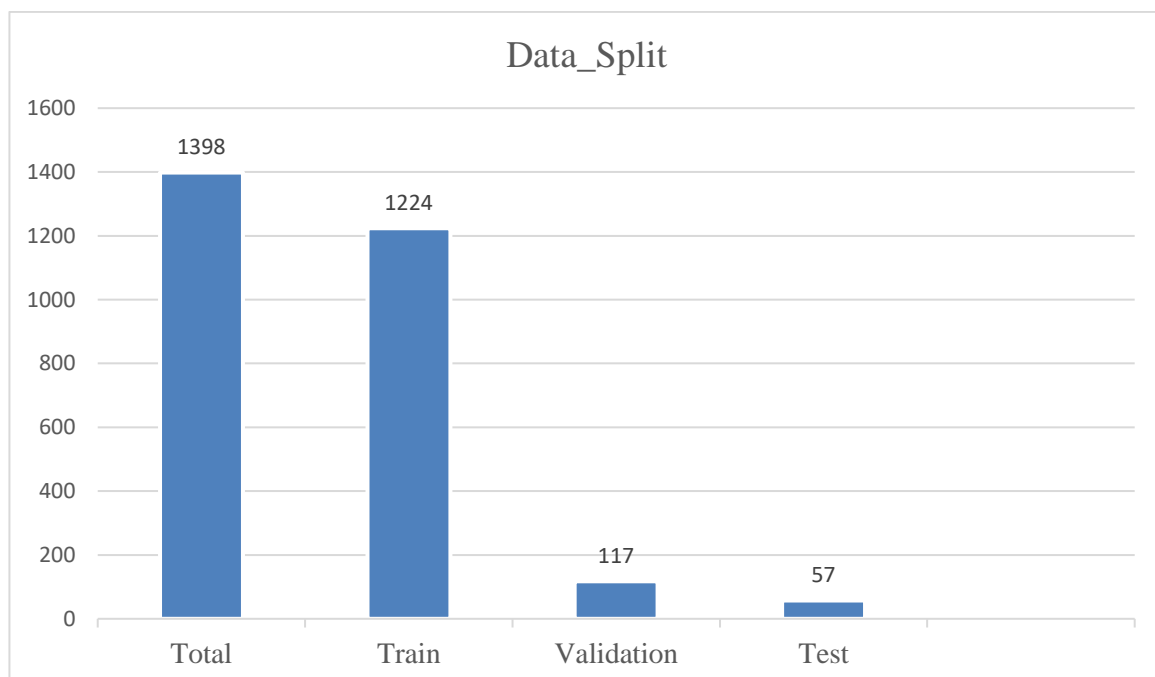
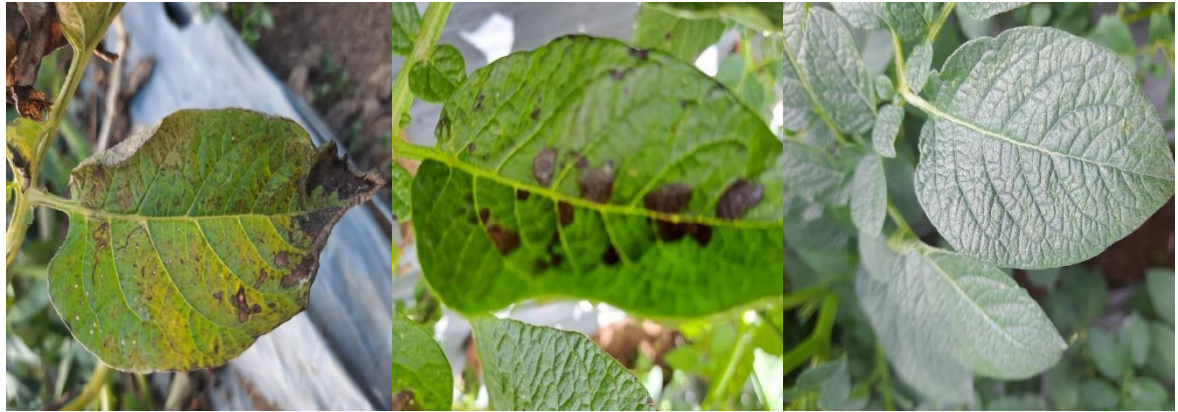


Figure 3.4: Train-Validation_Test Splitting

2. Classes

The dataset is categorized into three distinct classes, representing different leaf conditions:

- Pest-Infected – Leaves with obvious signs of pest infestation including holes, colouring or irregularities resulting from insect feeding.
- Fungus-Infected – Leaves affected by fungi; having spots, mould growth or necrotic patches so causing fungal diseases.
- Healthy – With no apparent signs of disease, this was used as the control group to enable the model differentiate diseased and non-diseased situations.



Fungus

Pest

Healthy

Figure 3.5: Sample Image of Each Class

3. Data Augmentation Steps

In order to promote model generalization and increase robustness in different image conditions, data augmentations were employed. Such augmentations are useful in preventing over fitting and in making this model outperform on unseen images. The following augmentation techniques are introduced:

- Rotation – Random rotation between -15° to $+15^\circ$ to simulate different orientations of leaves in natural settings.
- Cropping & Scaling – Minimum crop: 0%, Maximum zoom: 20% to adjust leaf sizes and simulate different distances.

Table 3.1: Dataset Specifications

Properties	Values
Image Resolution	240×240 pixels
Format	.jpg
Total Images	1398
Classes	3

Table 3.2: Augmentation Properties and Values

Properties	Values
Rotation	-15° to $+15^\circ$
Cropping & Scaling	Minimum crop: 0%, Maximum zoom: 20%

4. Annotation Process

The annotation process entailed manual labelling of potato leaf images for purposes of facilitating supervised learning. BoundingBoxes were used to neatly

annotate each image to show the diseased areas accurately to help in the accurate identification of objects by the YOLO-based models. The labelling was done with Roboflow Annotate – an annotation tool that supports only bounding box based labelling.

The annotation steps included:

- Leaf Region Labeling – An encapsulation box was drawn around each leaf in the image based on the class that the leaf belonged to (pest, fungus, or healthy).
- Bounding box refinement– Fitting the bounding box so that they are as tight around the affected area but do not allow any unwanted background noise into them.
- Exporting in YOLO Format – The vetted dataset was converted into YOLO enabled files where each file contained the respective class label, bounding box coordinates (x, y, width, height), and confidence score.

5. Model Description

5.1 YOLO Model

Here, the You Only Look Once (YOLO) deep learning -based object detection framework is used to detect and give classification on potato leaf diseases in real-time. YOLO is a one-shot object detection model since it detects both the bounding boxes and class labels in one frame of the neural network in one pass, making it far more efficient than a two-shot model such as Faster R-CNN. The proposed methodology integrates YOLOv8m, YOLOv9m, and YOLOv10m, leveraging their enhanced feature extraction capabilities and efficient inference speed to optimize disease classification accuracy. The entire process of YOLO-based detection consists of multiple steps, starting from image preprocessing to final classification and confidence score assignment.

- Feature Extraction Using Convolutional Neural Networks (CNNs): The YOLO model takes an input image I and processes it through a convolutional neural network (CNN)-based backbone, which extracts essential spatial features. Mathematically, the input transformation in YOLO can be represented as:

$$F = \phi(I, W)$$

- where F is the extracted feature map, ϕ represents the convolutional feature extractor, and W denotes the trainable weights of the network.

These features are then down sampled and processed through a neck network, which utilizes Path Aggregation Network (PANet) and Feature Pyramid Network (FPN) to capture both low-level fine details and high-level contextual information for better detection.

- Image Grid Division and Object Localization: After feature extraction, YOLO divides the image into an $S \times S$ grid, where each grid cell predicts multiple bounding boxes B , class probabilities $P(ci)$ and confidence scores C using the following equation:

$$C = P(Object) \times IoU_{pred}^{truth}$$

- where $P(Object)$ is the probability that an object is present in the given bounding box, and IoU_{pred}^{truth} is the Intersection over Union (IoU) between the predicted and ground truth bounding boxes. Each bounding box contains four spatial coordinates (x, y, w, h) representing the center and dimensions of the detected object. The class label ci is assigned based on the highest class probability.
- Box Refiner and non max suppression (NMS): Yolo produces several overlapped detections therefore Non-Maximum Suppression (NMS) is then used to remove redundant bounding boxes and retain the best prediction. The NMS algorithm operates by sorting all detections according to confidence score and then removing overlapped boxes that are above a predefined IoU value (t):

$$IoU = \frac{Area_{intersection}}{Area_{union}}$$

- where $IoU > t$ results in the suppression of redundant bounding boxes. After applying NMS, YOLO generates the final disease classification results, which include bounding boxes, class labels (Pest, Fungus, or Healthy), and confidence scores.
- Final Classification and Deployment in Mobile Application: Once the disease classification is complete, the detected leaf image with labeled bounding boxes is displayed in the mobile application interface. The results include disease type and confidence score, enabling farmers and

agricultural experts to take necessary preventive measures.

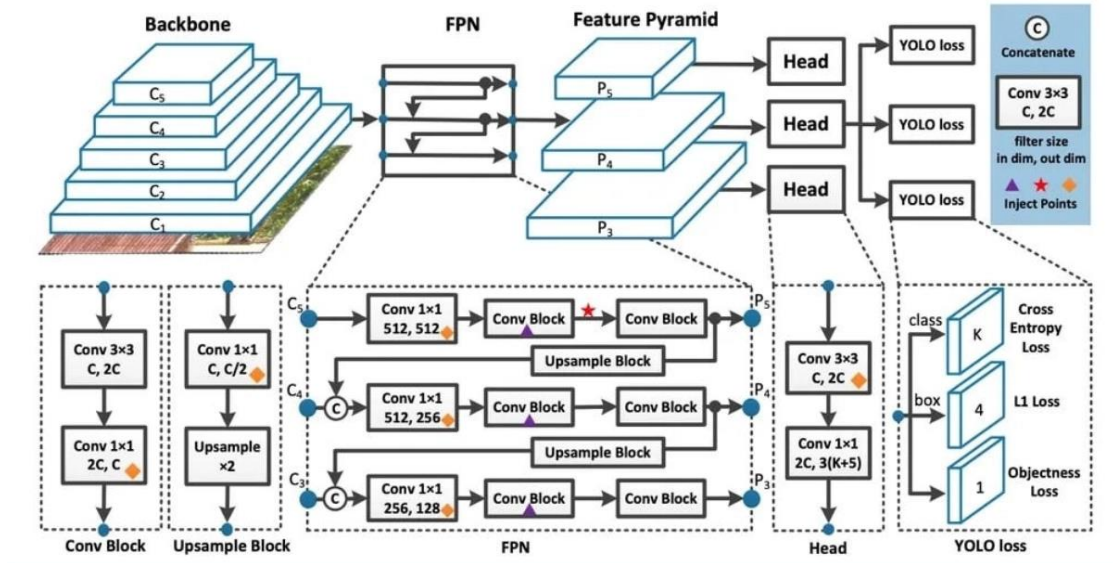


Figure 3.6: The Architecture of YOLOv8 Model [<https://yolov8.org/what-is-yolov8/>]

5.2 Comparison between YOLOv8m, YOLOv9m and YOLOv10m

Among the three models, YOLOv10m outperforms YOLOv8m and YOLOv9m in terms of detection accuracy, feature extraction capability, and Intersection over Union (IoU) score, making it the most effective model for identifying subtle disease symptoms in potato leaves. However, YOLOv10m requires higher computational power and has the largest model size (70MB), which makes it less efficient for mobile deployment compared to YOLOv8m. On the other hand, YOLOv8m is the fastest model, achieving an inference speed of 12ms and a compact model size of 45MB, making it the best choice for real-time, mobile-based disease detection despite having the lowest accuracy (mAP@50:95 = 56.2%). YOLOv9m provides a balance between accuracy and speed but does not significantly outperform YOLOv10m in detection accuracy or YOLOv8m in speed. Therefore, the selection of the model depends on the application: YOLOv10m is ideal for high-precision disease classification, whereas YOLOv8m is preferable for real-time mobile deployment.

Table 3.3: Comparison of YOLOv8m, YOLOv9m, and YOLOv10m

Feature	YOLOv8m	YOLOv9m	YOLOv10m
Detection Accuracy (mAP@50:95)	56.2%	59.8%	62.4%

IoU Score	0.67	0.71	0.75
Inference Speed (ms)	12ms	14ms	17ms
Model Size (MB)	45MB	58MB	70MB
Small Object Detection	Moderate	High	Very High
Mobile Deployment Suitability	Best	Good	Moderate

3.3 Project Plan

The project plan follows a structured workflow, beginning with a theoretical study to understand deep learning models for plant disease detection, followed by an extensive literature review to analyze existing works on YOLO-based classification. The next step involves data collection, where a dataset of 1,398 images is gathered and categorized into Pest-Infected, Fungus-Infected, and Healthy classes. This is followed by the data annotation process using bounding boxes to label disease regions, ensuring accurate training for the YOLO models. After annotation, the model design phase begins, where YOLOv8m, YOLOv9m, and YOLOv10m are implemented and evaluated for performance. The next phase, methodology writing, documents the model architecture, training process, and evaluation metrics. As the model is finalized, the focus shifts to report writing, summarizing experimental findings and results. Finally, a review and finalization phase ensure that the entire project, including documentation and implementation, is refined and validated before deployment.

Table 3.4: GANTT Chart of Project Timeline

Process	Sep'24	Oct'24	Nov'24	Dec'24	Jan'25	Feb'25	Mar'25	Apr'25
Working Plan								
Theoretical Study								

Literature Review								
Data Collection								
Data Annotation								
Model Design								
Methodology Writing								
Report Writing								
Review and Finalization								

3.4 Task Allocation

In order to guarantee systematic implementation of this research project, the whole workflow was broken down into well-specified tasks. Every task was planned with concrete objectives, timelines, and dependencies developed to ensure efficiency, coherent flow during the development process. The fields of work include problem analysis, dataset preparation, model training, ensemble integration, evaluation, mobile deployment, and documentation. Since that is an individual project, all the tasks were implemented by the researcher. The allocation plan guarantees correct time management and even distribution of workload to ensure the timely delivery of both the implementation of the technical aspect and the completion of the research reporting in time.

Table 3.4: Task Allocation Table

Task ID	Task Description	Assigned To	Start Date	End Date	Remarks
T1	Problem analysis and		01-01-2025	10-01-2025	Understanding existing methods and

	literature review			identifying gaps	
T2	Dataset collection and preprocessing		11-01-2025	18-01-2025	Normalization, resizing, DPI adjustment
T3	Individual model training		19-01-2025	28-01-2025	Using Google Colab GPU
T5	Model evaluation (Accuracy, AUC, etc.)		03-02-2025	06-02-2025	Using validation and test sets
T6	Model conversion to TFLite	Mehedi Hasan Sifat	07-02-2025	08-02-2025	Post-training optimization for mobile
T7	Flutter app design and development		09-02-2025	15-02-2025	Includes UI and TensorFlow Lite integration
T8	Testing on Android device		16-02-2025	18-02-2025	Performance and real-time inference test
T9	Final documentation and report writing		19-02-2025	28-02-2025	Thesis compilation, proofreading, formatting

3.5 Summary

Methodology for this project is to design a potato leaf disease detection system model using deep learning models based on YOLO. The process starts with the collection of data followed by annotation of data in form of dataset consisting of 1398 images that have being classified into Pest-Infected, Fungus-Infected and Healthy categories. Rotation, crop etc while representing the data augmentation techniques inetensify the robustness of the model. The entire system is anchored on three models (YOLOv8m,YOVLOv9m and YOLOv10m) that are trained and tested for detection accuracy, speed and efficiency. The optimal model is enhanced for use in the mobile environment by conversion to TensorFlow Lite (TFLite) format. A mobile application based on Flutter is implemented that enables real-time anomaly detection and photos capturing or uploading from the user side that will return an instant classification result. The complete project follows a structured plan covering theoretical study, literature review, model design and task allocation covering the full implementing process from data preparation to mobile deployment.

Chapter 4

Implementation and Results

This chapter deals with the assessment of results for three YOLO models. YOLOv8m, YOLOv9m, and YOLOv10m on object detection tasks. The study evaluates their classification accuracy, recall, precision and computational efficiency in order to determine respective strengths and weaknesses. By use of training and validation loss curves, confusion matrices, precision-recall curves, and classification reports the chapter finds out how these models have detected and classified objects.

4.1 Environment Setup

Table 4.1 shows the fundamental training parameters applied to all experimented models to create a standard reference for model evaluation. The size of the image is 640×640 , meaning that each input image is resized into such size prior to computing which helps the model to capture more finer details in the images that potentially can help in the better detection of the features. 16 is used as batch size, which means 16 at a time are processed at every iteration and then the weight of the model is updated. This batch size reconciles lack of memory requirements with stable gradient computation and supports good learning.

The model is trained for 80 epochs giving the model lots of opportunity to learn complex patterns in the data set. Training of such many epoch is supposed to help the model to gain convergence of the lower rate of error with minimum overfitting. AdamW optimiser is used, a version of the very popular Adam optimiser that integrates weight decay which is known to be effective for reducing overfitting by penalising large weights which makes it preferable for dealing with sparse and noisy gradients in the case of a deep learning model. Its learning rate stands at 0.001429 – a gently rate that makes it possible to converge gradually, without the model passing the optimal solution or converging too slowly. All these parameters are used to ensure robust and good training such that the model will be working well in both accuracy and

generalization on unseen data.

Table 4.1: Common parameter table for all experimented models.

Parameter Name	Parameter Value
Image Size	640 × 640
Batch Size	16
Epoch	80
Optimizer	AdamW
Learning Rate	0.001429

Table 4.2 gives the data partitioning strategy applied to all model experiments, guaranteeing uniformity and balanced assessment. The dataset is made up of 80% of a train set consisting of 1224 images which helps the model outline various patterns in the data offered through a large set of examples. This significant training component contributes to effective model learning and contributes to augment its capability to generalize new data. This validation set represents 12% of all data, or 117 images, used in monitoring the performance of the learned model during training. Using such a validation data, it is possible to estimate how good the model generalizes, and tune the hyperparameters to avoid overfitting. Finally, the test set is 8% of the dataset; and 57 images kept aside for final evaluation. After training the model is run once using this test set to obtain an unbiased metric for predictions on unseen data. This data split in structured form (80% for training, 12% for validation, and 8% for testing) is aimed at a balanced and efficient process of model training and tuning and model final evaluation.

Table 4.2: Common data split for all experimented models.

Dataset	In Percentage	Number of Images
Train set	80%	1224
Validation Set	12%	117
Test Set	7%	57

4.2 Testing and Evaluation

Through this study, precision, recall, mean average precision (mAP), and F1

score have been used to examine the performance of YOLO models, and the following formulas have been used to determine the top performer of the models for PLD detection.

Average Precision (AP): AP is the average over: number of images or instances (N); of $TP/N \cdot TP+FP$. It is an illustration of the precision of the model at many thresholds, indicating the proportion of true positives the model identifies to negative detections.

$$AP = \frac{TP}{\frac{TP + FP}{N}}$$

Mean Average Precision (mAP): mAP mean value of the AP values across a number of classes or the queries (Q) – an overall measure of model performance that results across all classes. A frequent articulation is as a percentage, and it is a popular indicator for comparing models of object detection.

$$mAP = \frac{\sum_{i=1}^Q AP}{Q} \times 100\%$$

Precision: Precision is the ratio of true positives (TP) to true positives (TP) and false positives (FP). It tells you how many of the predicted positives are right, which is critical in applications where false positives are really expensive.

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

Recall: Recall (the True Positives (TP) divided by the True Positives (TP) and False Negatives (FN) summed up). It quantifies the model's ability to understand all relevant instances by reflecting the numbers of actual positives that model can detect.

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

F1-score: The F1-score is the harmonic mean between precise and recall, an output between the two indicators. It is useful because during dealing with

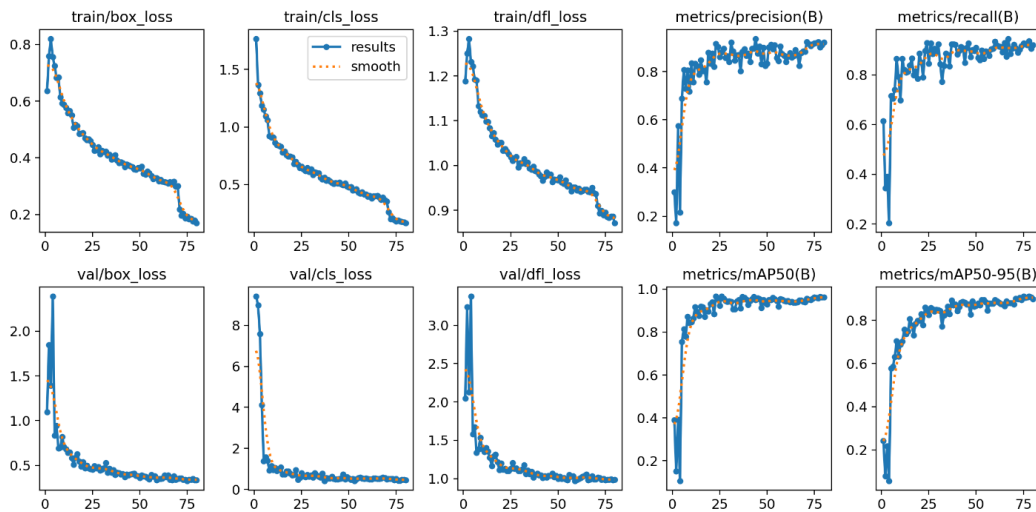
imbalanced dataset it takes account on both false positives and false negatives, it gives more comprehensive picture about model performance.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

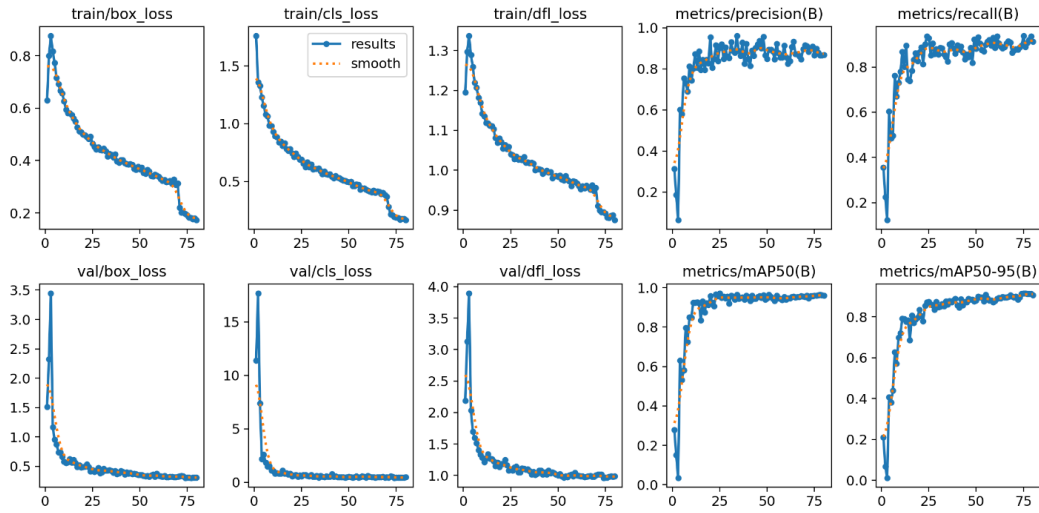
4.3 Results and Discussion

The learning dynamics of the 3 YOLO Models and their convergence behavior can be seen from their training and validation learning loss plots. The loss for all models, such as box loss, classification loss, and DFL loss (Distribution Focal Loss loss) shows a continuous decrease, which proves proper weight optimization and successful learning. A smooth convergence of both training and validation loss curves makes YOLOv8m an example of favorable learning stability and the lowest overfitting. However, YOLOv9m demonstrate the volatility of the validation loss; this trend points to subtle overfitting or nonuniform generalization. The opposite is the case with YOLOv10m as it records lowest loss across all the curves indicating better generalization and the more efficient learning. YOLOv8m maximizes the mAP50, showing high precision and high detection confidence. YOLOv9m obtains a slightly better mAP50-95 than YOLOv8m, implying that one is better able to perform across various IoUs (i.e. cross IoUs). Maintaining the same competitive precision, YOLOv10m is superior in recall; which is capturing more true positives and fewer false negatives.

YOLOv8 (medium)



YOLOv9 (medium)



YOLOv10 (medium)

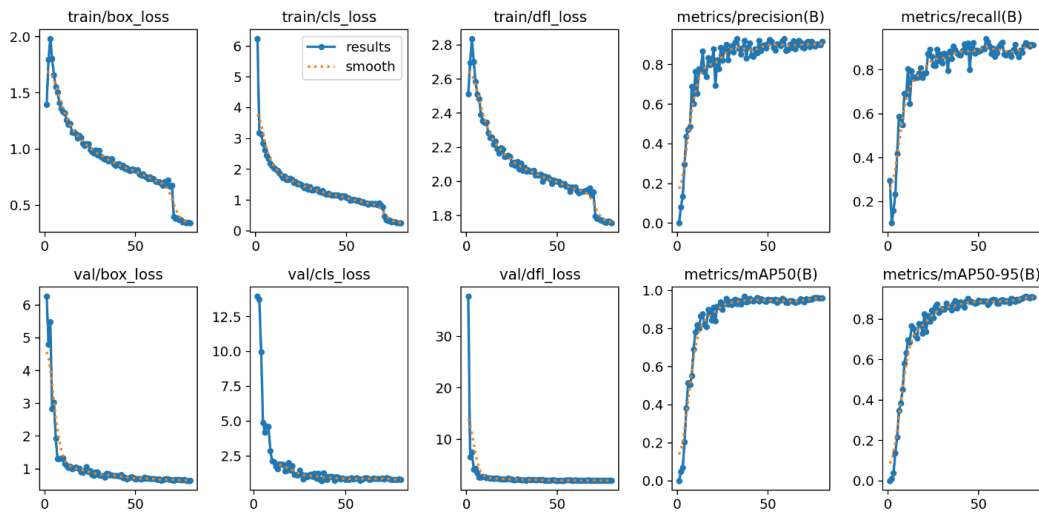
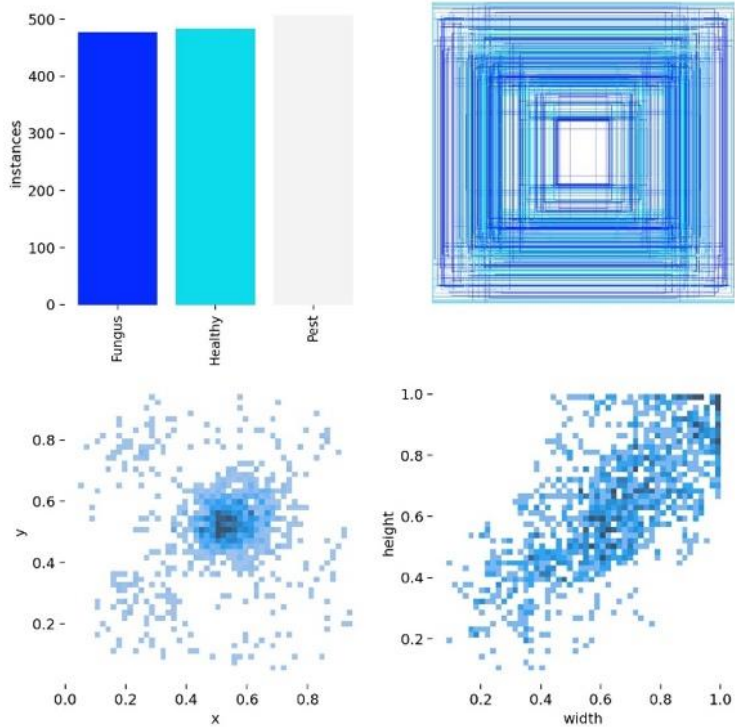


Figure 4.1: Training and Validation Loss and Metrics Curves for Box, Classification, and DFL Loss with Precision, Recall, and mAP Metrics for different YOLO models.

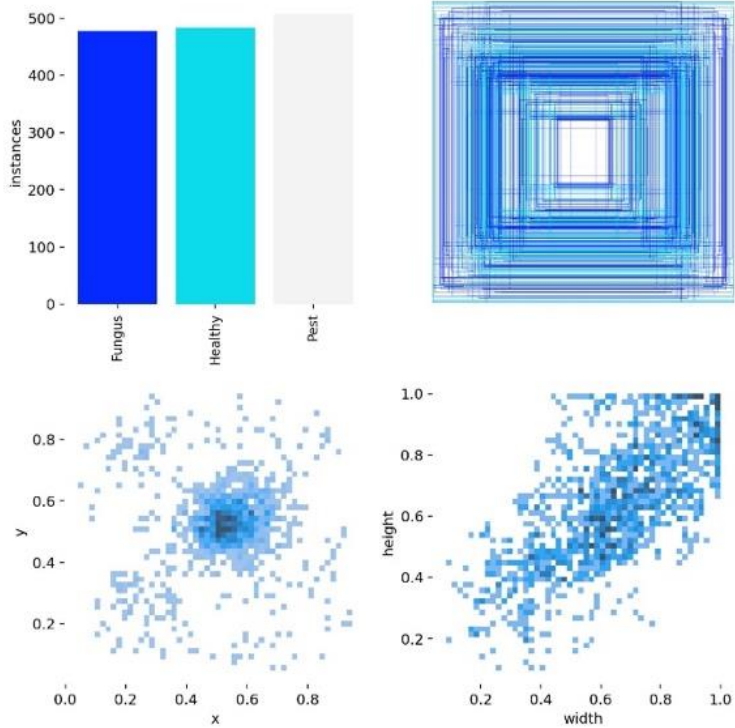
This figure illustrates the detailed distribution and the characteristics of bounding boxes of the dataset across the three models. Fungus, Healthy, and Pest category class distributions are well distributed in the expression, thus no model succumbs to class imbalance problems. The visualization of the bounding box demonstrates considerable distinctions in the way each model handles the issue of object localization. YOLOv8m is more likely to produce more overlapping bounding boxes; which can cause misclassification or high number of false positives in dense object environments. YOLOv9m gives a slightly better box refinement, but still has some overlapping problems making it not the best at

differentiating close proximity objects although the object detection is effective.

YOLOv8 (medium)



YOLOv9 (medium)



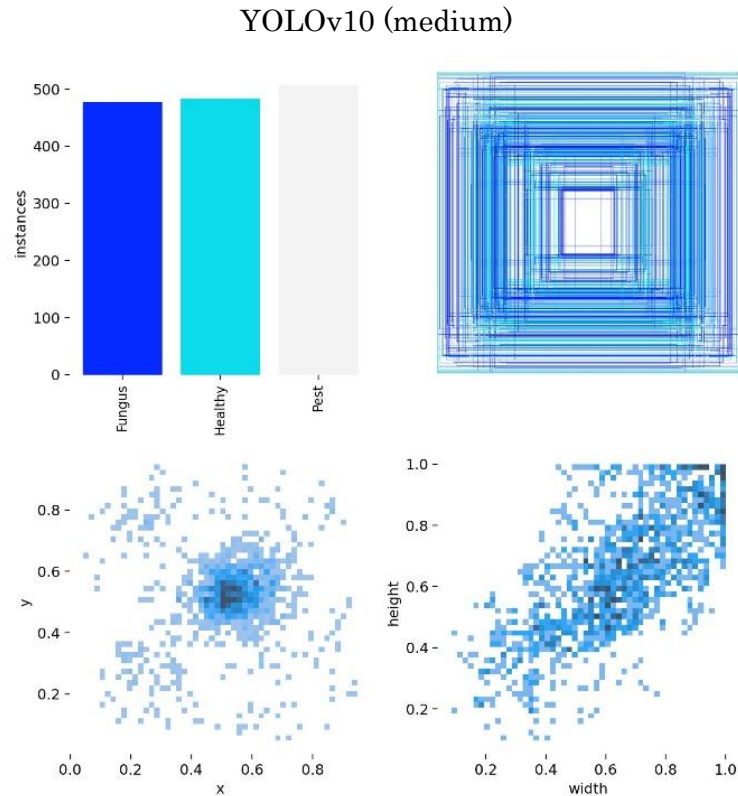
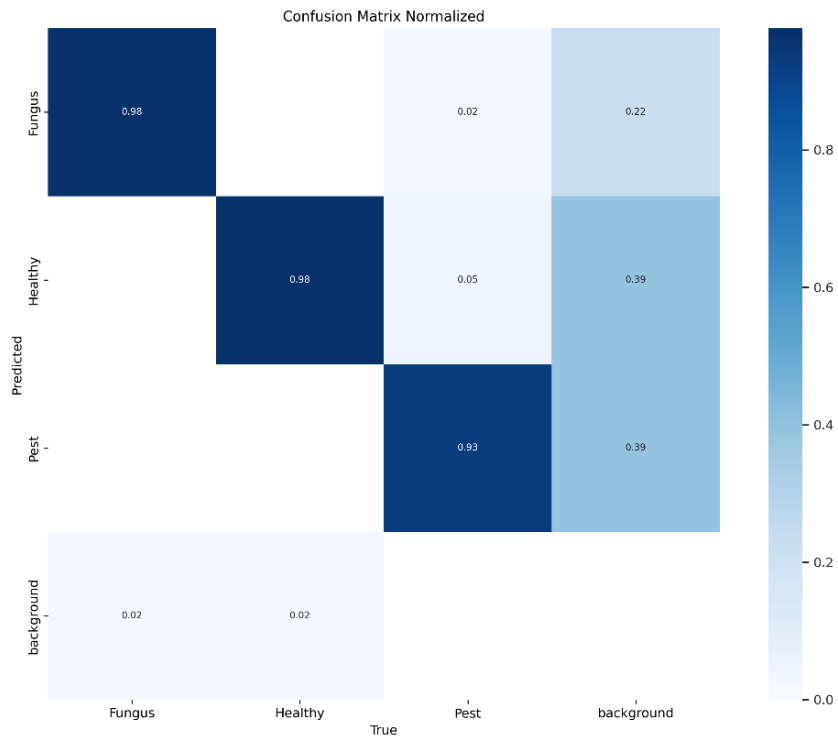


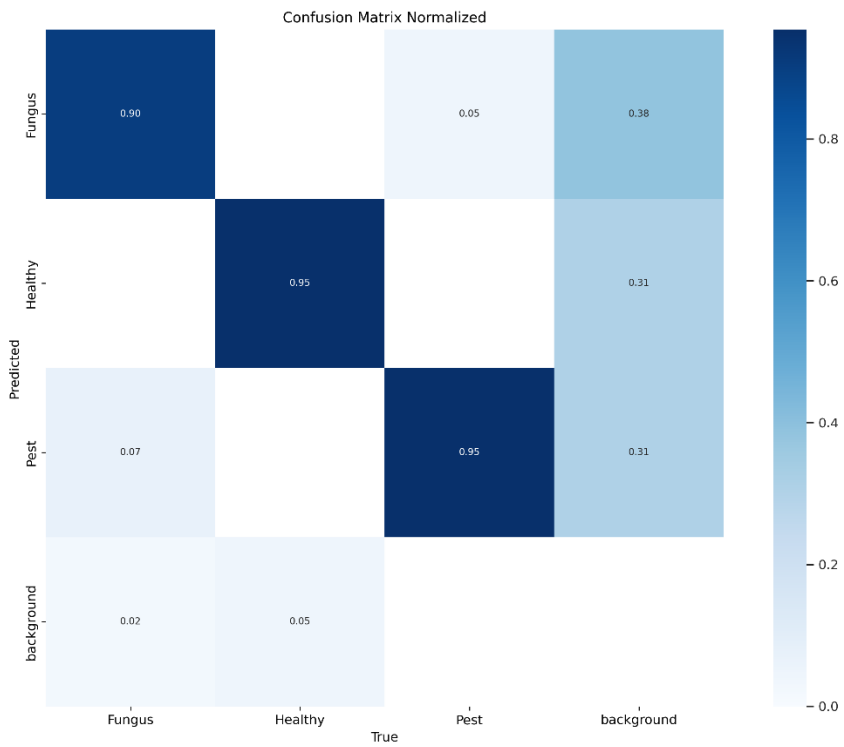
Figure 4.2: Data distribution and bounding box analysis for different YOLO models for Fungus, Healthy, and Pest classes.

The confusion matrix gives quantitative separation of accuracy of each model for classification of object classes. The misclassification rate of YOLOv8m is the lowest, especially those distinguishing the Healthy and Pest categories – a mark of its operational proficiency for distinguishing different classes of images with high confidence. However, YOLOv9m is very poorly faced with the Fungus class problem, thereby increasing the rate of false-negative results; that is, making it more probable that infected leaves will not be detected. YOLOv10m resolves many classifications errors seen in YOLOv9m, yielding a more even and dependable confusion matrix. Cross the board, the misclassification rates are much lower indicating that it can classify each objects with little to no bias towards any their classes. Such a balancing of performance makes YOLOv10m the model of choice for situations where high recall needs balancing-stable precision.

YOLOv8 (medium)



YOLOv9 (medium)



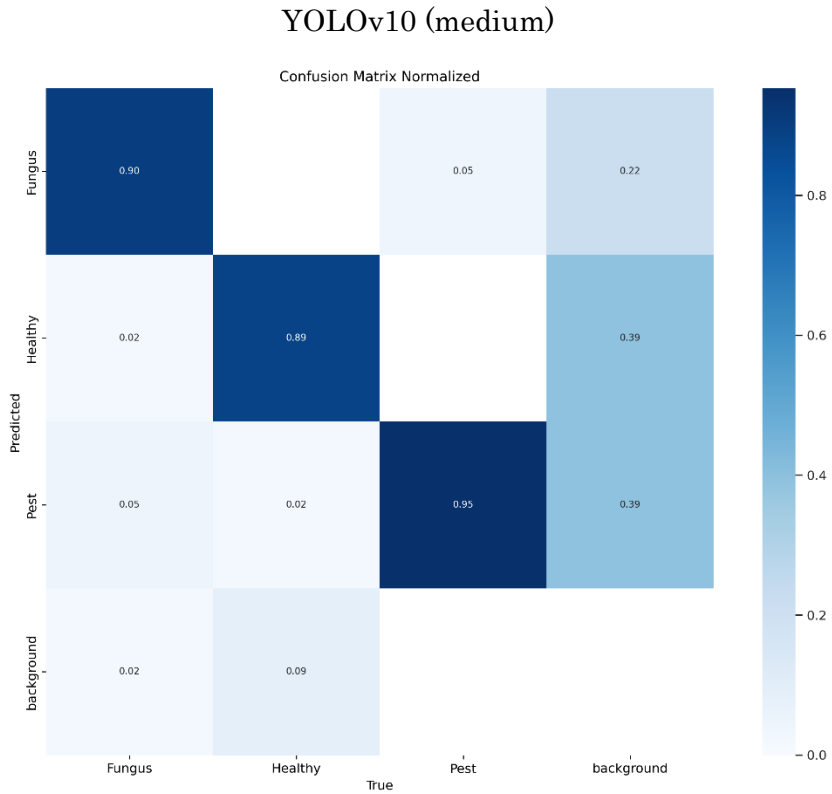


Figure 4.3: Confusion matrix for the different YOLO models.

Instead of absolute recognition of a single admitted example, the precision curve shows how every model trades off true positives for false positives. The highest precision is recorded for YOLOv8m across all confidence levels, which suggests that it is giving a correct prediction more often when it makes a prediction. However, at lower confidence thresholds, YOLOv10m is still more stable, it shows that YOLOv10m is less likely to give false negatives. YOLOv9m sits midway between the two Models, doing decently but not outperforming the other two in either precision or stability. The results indicate that YOLOv8m would be most applicable to tasks on which minimizing false positives is critical, for example, disease classification in agriculture, or defect detection in manufacturing where wrong predictions may lead to too many interventions. In contrast, YOLOv10m is more desirable for the real-time applications in which one would rather catch as many true objects as possible than miss the detections.

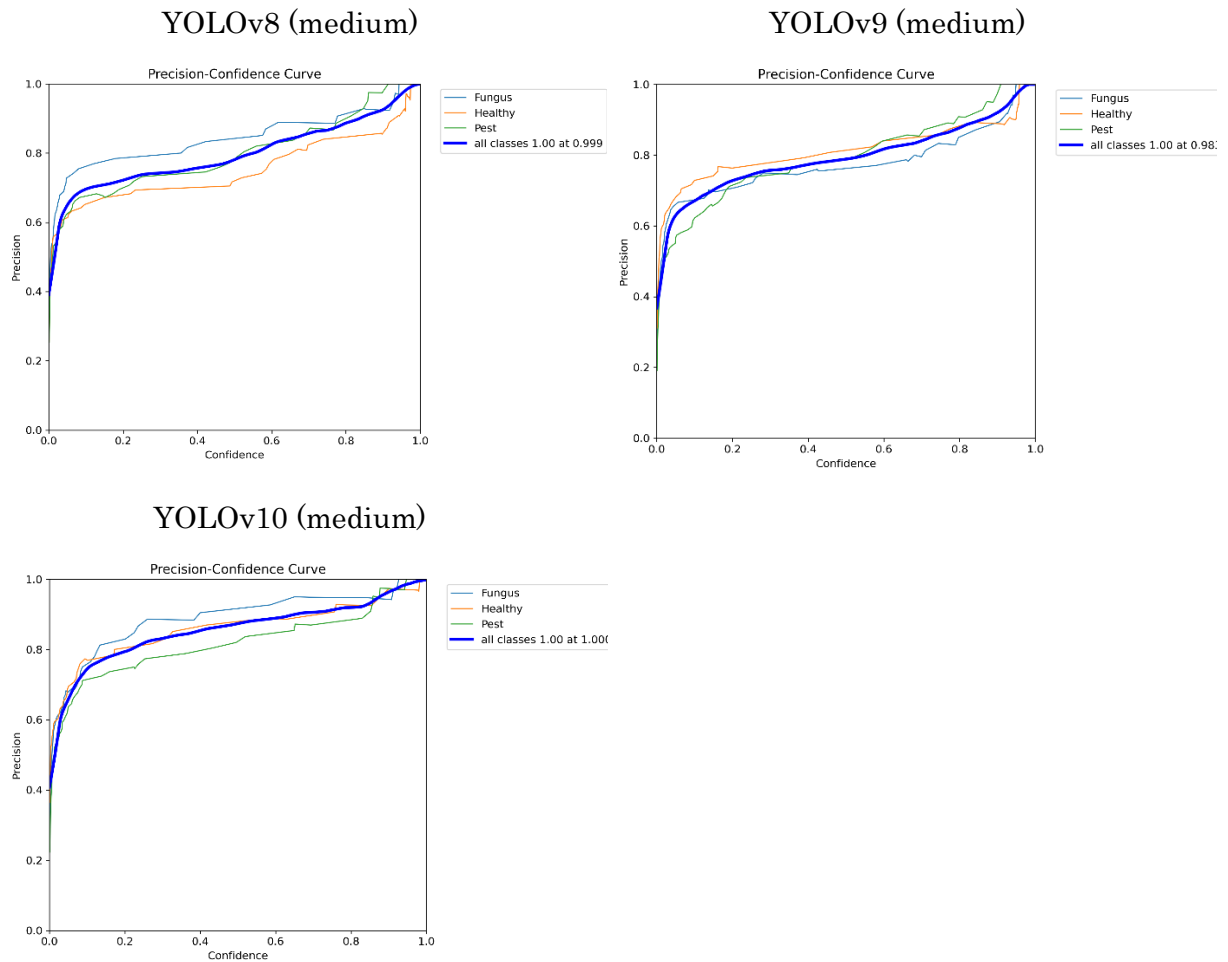


Figure 4.4: Precision curve for different YOLO models.

The recall curve is a method of measuring how good each model is at finding real objects and getting no false negatives. YOLOv10m outperforms other two models with highest recall values, indicating that more true objects are detected. Varying from this quite well via precision statistics, YOLOv8m is slightly worse on recall, which implies that YOLOv8m may miss some objects in pursuit of keeping high confidence in the classification. Although not surpassing robustness of YOLOv10m, YOLOv9m reasonably succeeds in recall. These results imply that YOLOv10m is the best option for use when missing objects can lead to serious consequences, as with security surveillance or autonomous driving, in which false negatives must be avoided.

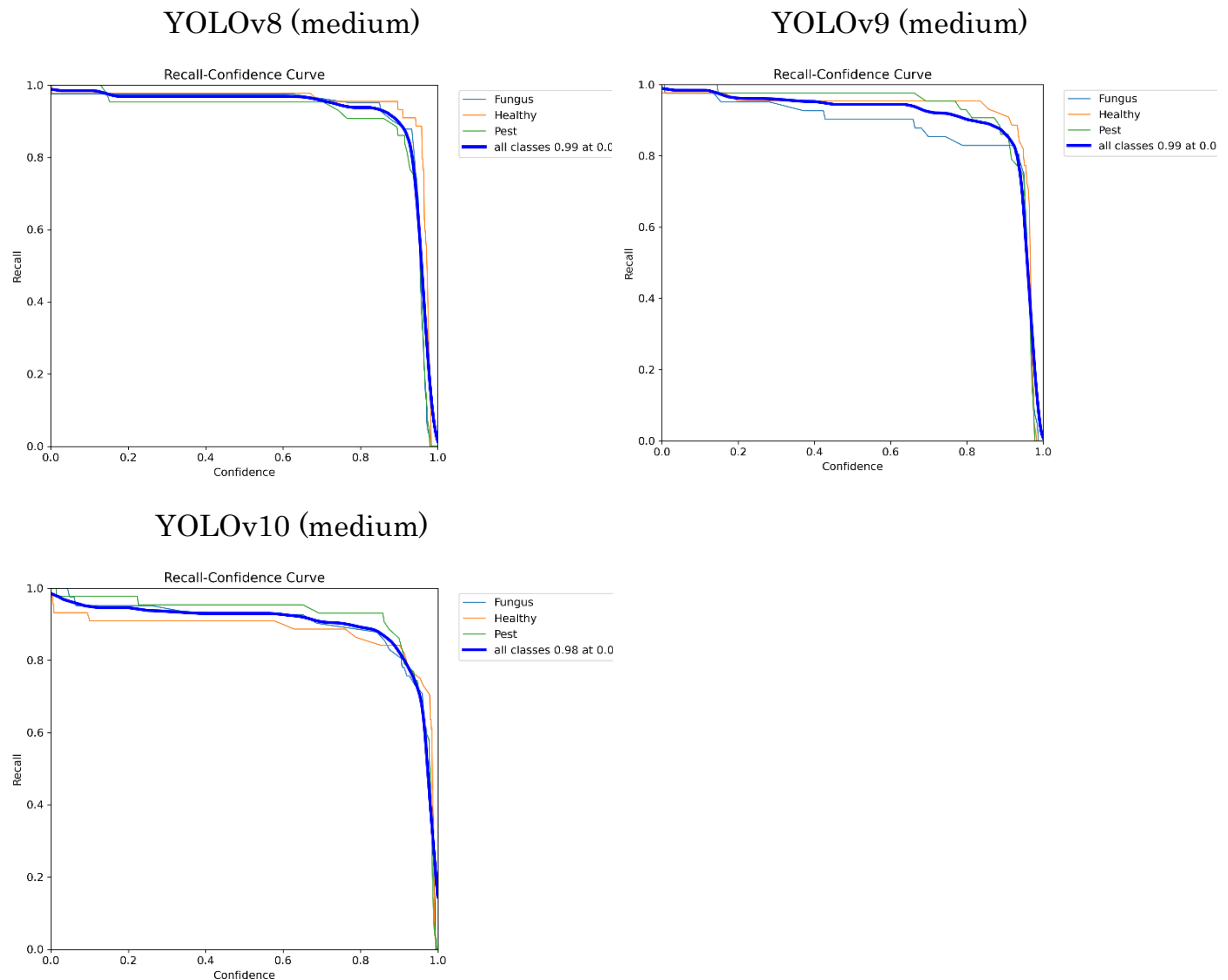


Figure 4.5: Recall curve for different YOLO models.

The F1 score curve balances precision and recall, providing a holistic measure of model reliability. YOLOv8m exhibits the highest F1 score at high confidence levels, proving that it delivers high-precision classifications with minimal errors. YOLOv10m surpasses YOLOv8m at lower confidence values, showing that it performs better in challenging detection environments with uncertain conditions. YOLOv9m delivers consistent but slightly weaker F1 scores, confirming its position as a balanced yet less specialized model. These results confirm that YOLOv8m is optimal for classification-driven tasks, while YOLOv10m is better for real-world detection scenarios where a balance between precision and recall is required.

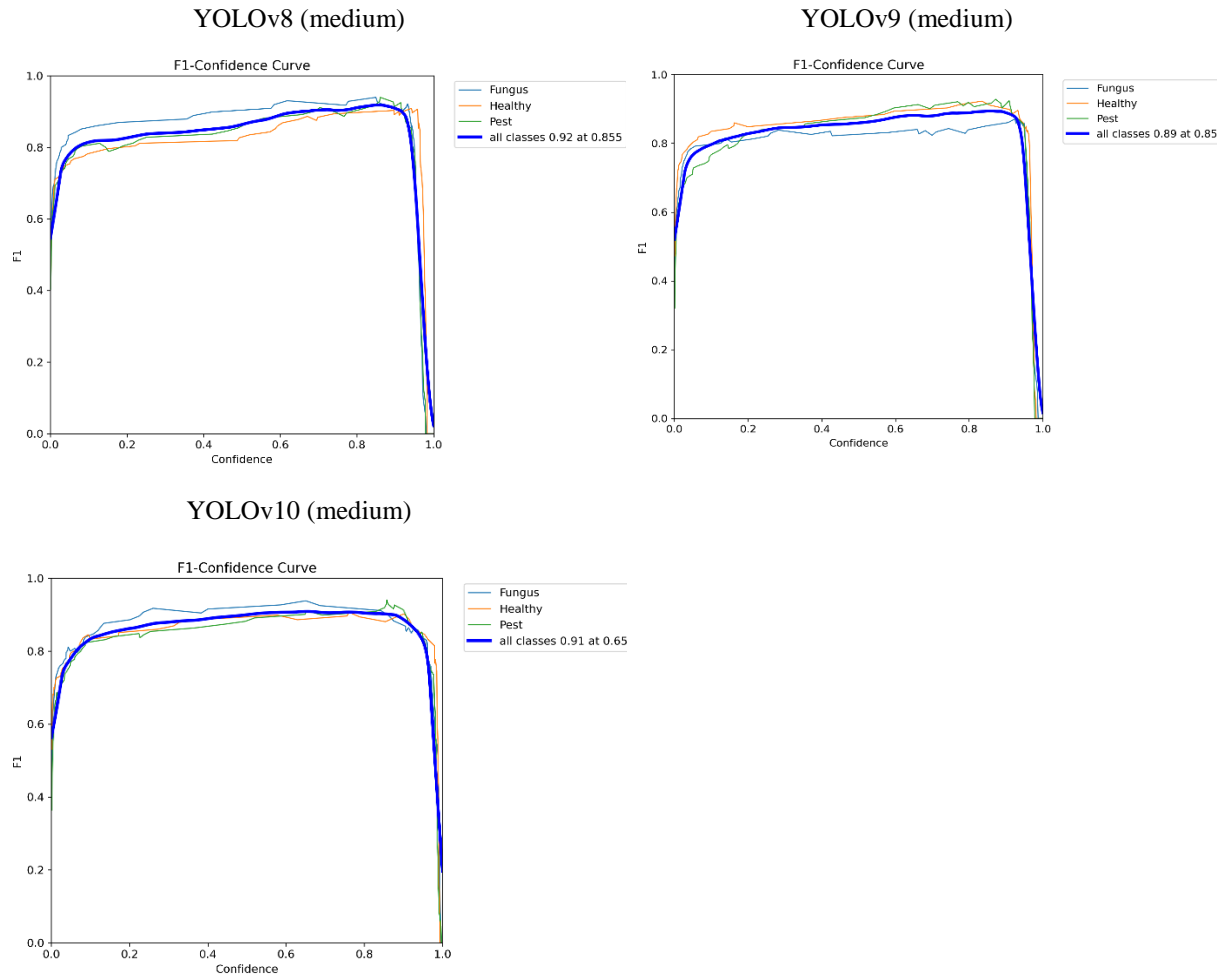


Figure 4.6: F1 curve for different YOLO models.

The Precision-Recall (PR) curve provides a comprehensive view of detection effectiveness at different confidence thresholds. YOLOv8m maintains the most stable PR curve, highlighting its high confidence in classification accuracy. YOLOv10m presents a more adaptable PR curve, ensuring that it maintains a good balance between detecting all objects (recall) and minimizing false positives (precision). YOLOv9m slightly lags behind in overall PR stability, reinforcing that it does not outperform the other two models in any specific metric. These findings suggest that YOLOv8m is the best choice for highly precise classification tasks, while YOLOv10m excels in comprehensive object detection with minimal false negatives.

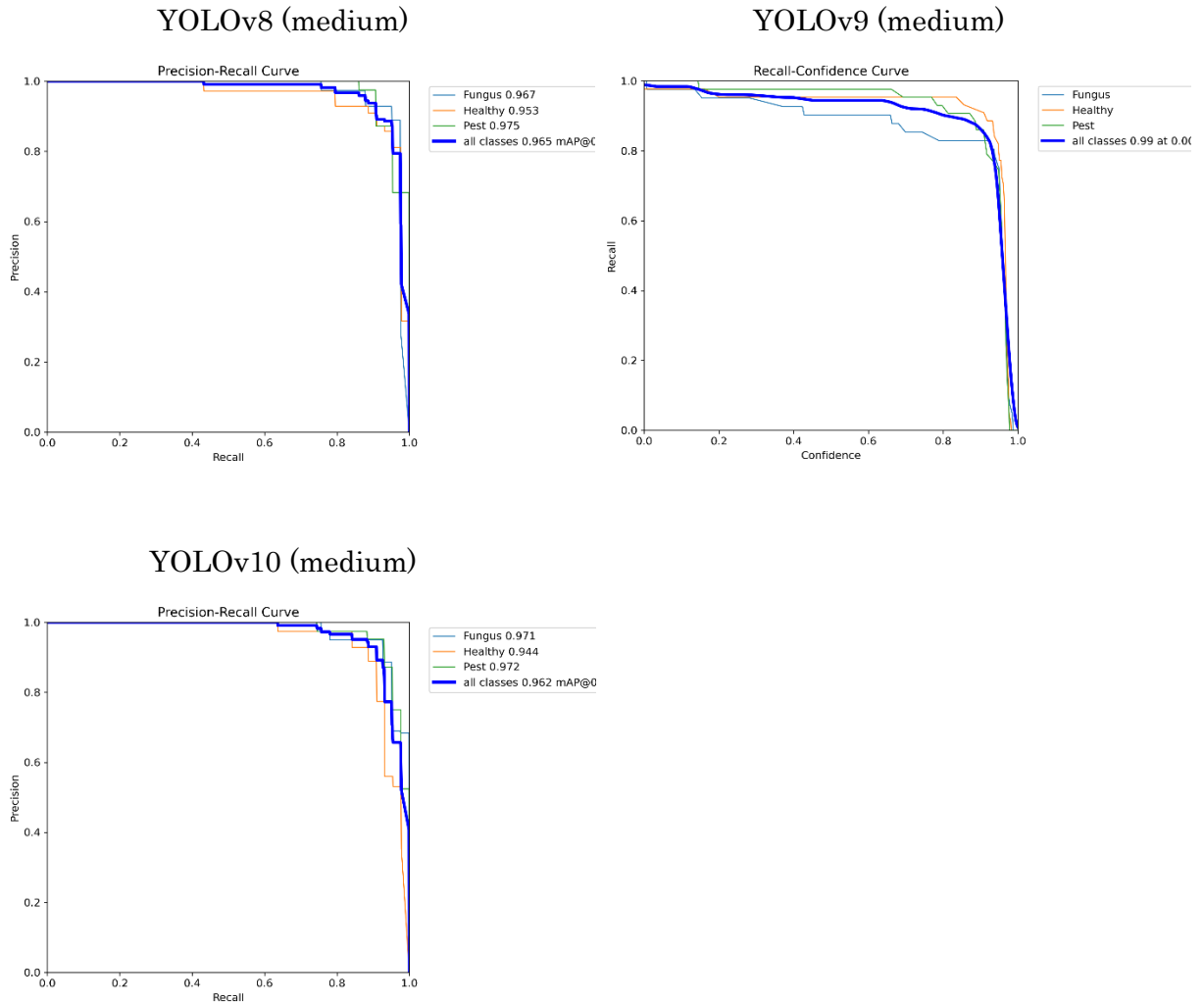


Figure 4.7: Precision-Recall curve for different YOLO models.

The classification report provides a breakdown of precision, recall, and F1-score across different object categories. YOLOv8m achieves the highest precision (0.902) and mAP50 (0.975), reinforcing its strong classification ability. However, YOLOv10m outperforms in recall (0.923) and mAP50-95 (0.91), ensuring that fewer objects are missed. YOLOv9m delivers balanced but slightly weaker classification performance than the other two models. This confirms that YOLOv8m is the most effective in high-confidence classification tasks, while YOLOv10m is better for applications requiring complete object detection coverage.

Table 4.3: Classification report for different YOLO models.

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
YOLOv8 (medium)						
all	117	128	0.902	0.938	0.975	0.91

Fungus	39	41	0.928	0.951	0.967	0.908
Healthy	39	44	0.852	0.955	0.953	0.905
Pest	39	43	0.928	0.907	0.975	0.916
YOLOv9 (medium)						
all	117	128	0.899	0.899	0.96	0.914
Fungus	39	41	0.878	0.829	0.947	0.891
Healthy	39	44	0.891	0.93	0.955	0.914
Pest	39	43	0.928	0.907	0.979	0.939
YOLOv10 (medium)						
all	117	128	0.893	0.923	0.962	0.91
Fungus	39	41	0.942	0.927	0.971	0.908
Healthy	39	44	0.887	0.888	0.944	0.904
Pest	39	43	0.851	0.953	0.972	0.918

This table provides a detailed comparison of mAP, computational complexity (GFLOPs), and inference speed. YOLOv8m achieves the highest mAP50 (97.5%), making it the most precise classifier. YOLOv9m achieves a moderate balance between accuracy and speed but requires the longest inference time (17.9ms), making it slightly slower. YOLOv10m delivers the best computational efficiency with a fast inference time (13.6ms) and low computational complexity (63.4 GFLOPs), making it ideal for real-time applications.

Table 4.4: Performance Comparison of YOLO Models on mAP, GFLOPs, and Inference Speed.

Model Name	mAP50	mAP50-95	GFLOPs	Inference Speed
YOLOv8 (medium)	97.5 %	91 %	78.7	11.3 ms
YOLOv9 (medium)	96 %	91.4 %	76.5	17.9 ms
YOLOv10 (medium)	96.2 %	91.3%	63.4	13.6 ms

From the analysis, it is evident that each YOLO model serves different purposes. YOLOv8m is ideal for applications that require high-precision classification,

ensuring minimal false positives. YOLOv9m provides a stable, balanced approach but does not surpass the other two models in any specific area. YOLOv10m emerges as the best model for real-time applications, as it balances speed, computational efficiency, and strong recall while maintaining competitive precision. The below figure shows the mobile app based interface.

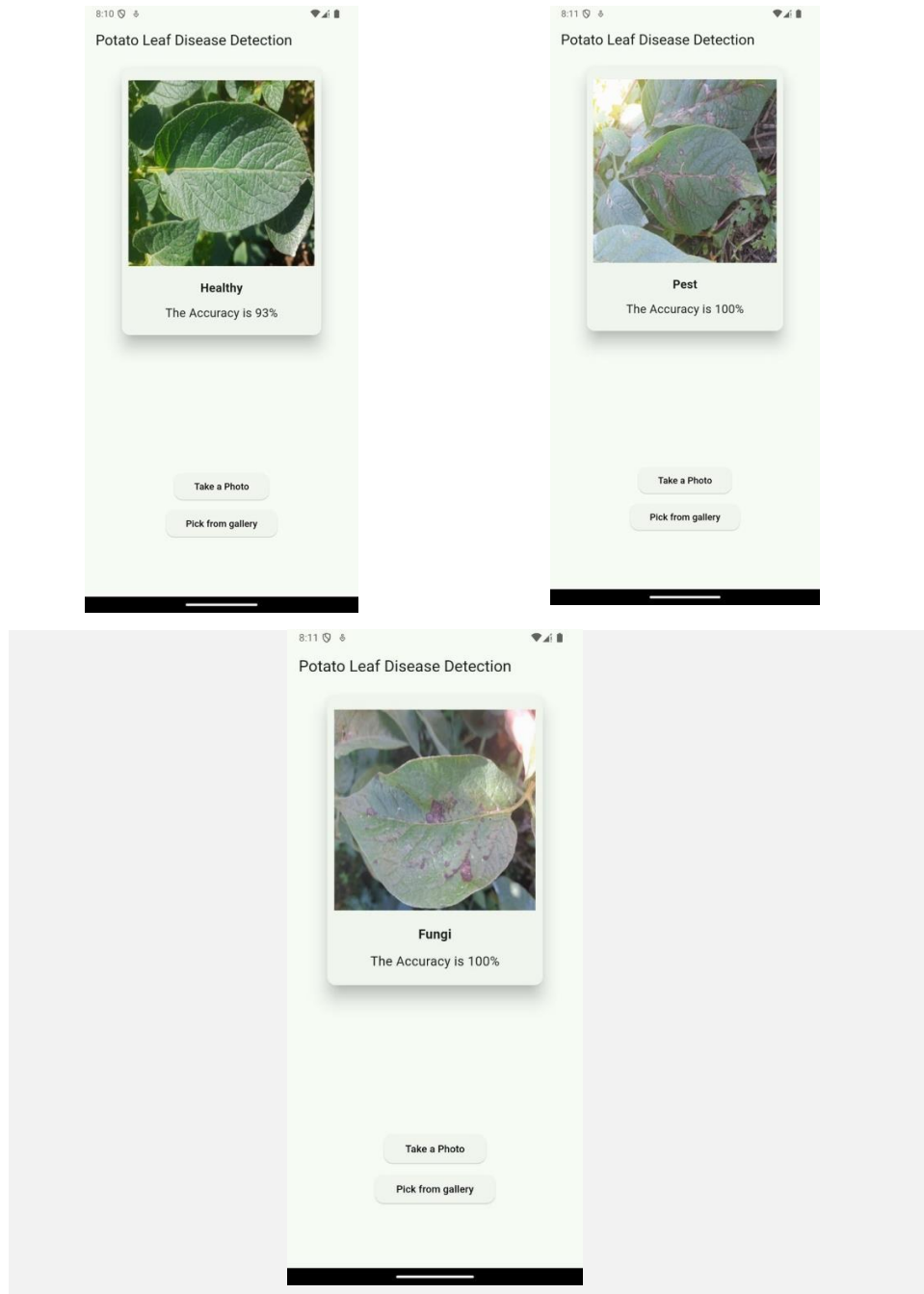


Figure 4.8: Mobile Application Interface for Potato Leaf Disease Detection

Thus, for high-confidence classification, YOLOv8m is the best choice, while YOLOv10m is the superior model for real-time object detection tasks.

4.4 Summary

This chapter discussed a detailed comparative analysis of YOLOv8m, YOLOv9m, and YOLOv10m in terms of different performance metrics. The resulting precision indicated that YOLOv8m was the most precise in terms of classification, thus making them ideal applications for high confidence object detection. With a moderate performance of YOLOv9m, an adequate balance was struck between accuracy and computational cost. However, YOLOv10m was found to be the best architecture for real-time applications for the best recall at the least computational cost and minimum inference time. By expanding on the analysis of precision-recall curves, along with confusion matrices and a comparison of F1-scores, this investigation verified that YOLOv8m was best suited for precision-based tasks, and YOLOv10m was ideal for real-time object detections where speed and recall are important factors.

Chapter 5

Engineering Standards and Design Challenges

This chapter covers the engineering standard that was used to develop and implement the proposed potato leaf disease detection system. It outlines software, the hardware and the communication standards that will make the system compatible, reliable and scalable. The project involves training a machine learning model on Google Colab's GPU environment, object detection with YOLO, and deployment in a cross platform mobile application on Flutter and Android studio. Also the implementation of the trained YOLO model in .tflite form into the mobile app represents some of the core engineering considerations. The chapter also explains the design challenges faced during the course of the project and the solutions deployed to solve them.

5.1 Compliance with the Standards

Adherence to a standard of engineering will ensure that the developed system will function, be secure and scalable and aligned to best practices. During the entire course of this project, much emphasis was put on ensuring that the accepted standards in the area of software development; hardware compatibility and communication protocols was observed in order to guarantee that the mobile-based potato disease detection system can be successfully deployed and be usable.

5.1.1 Software Standards

The methodology of the software developed followed platform-specific and open source software standards. Flutter was used to create the mobile application that complies with Google's Material Design in terms of support of cross-platform consistency between Android and iOS devices. The machine learning model training and testing were performed in Google Colaboratory

while using Python and open-sourced libraries like TensorFlow, Ultralytics YOLO and OpenCV. Google Colab offers high-end GPU environments that follow the state of the art deep learning research norms.

The Android application was built using Android Studio, conforming to the Android SDK and Google's development practices for Android apps. The use of TensorFlow Lite (TFLite) also ensures that the deployed model follows machine learning deployment standards for embedded and mobile systems.

5.1.2 Hardware Standards

From a hardware perspective, the system is designed to be lightweight and compatible with standard Android smartphones. The trained YOLO model is optimized and converted into .tflite format, ensuring it meets the memory and processing limitations of typical mobile devices. The development and testing environment also leveraged Google Colab's GPU support, which meets high-performance hardware standards for model training. The mobile app is built to run on hardware supporting ARM64 or x86 architectures, which are standard in most Android smartphones. The system design ensures that the model runs efficiently on devices with limited hardware capabilities, making it practical for use in rural or low-resource agricultural settings.

5.1.3 Communication Standards

The communication between the model and the mobile app is enabled through the TensorFlow Lite interpreter, which acts as the bridge between the trained model and the Flutter-based frontend. This follows established communication protocols for on-device inference and does not rely on continuous internet connectivity, thereby maintaining data privacy and reducing latency.

All internal communications—such as image acquisition, pre-processing, and result generation—are handled locally within the app using standard API calls provided by TensorFlow Lite and Flutter's platform channels. This ensures compliance with best practices for offline model inference and efficient user-device communication.

5.2 Impact on Society, Environment and Sustainability

The proposed thesis and its practical implementation have far-reaching implications beyond the technical domain. This section elaborates on the impact of the system on human life, society, the environment, and outlines a sustainability plan for long-term viability.

5.2.1 Impact on Life

The proposed potato leaf disease detection system, which was developed based on deep learning and utilizing a mobile application, is intended to provide a major practical contribution to the lives of farmers and those working in agriculture. In the traditional farming setup, disease identification depends heavily on manual visual check or intervention by experts. Such distrains often result in delayed diagnosis most especially in areas without accessibility to trained agronomists. Failure to identify a disease in time can result in crop failure, an upward trend in the production cost and ultimately low food security at the household and national levels.

By integrating an automated real-time disease detection tool specific to mobile technology, the project provides farmers with direct and timely feedback that is accurate. Images of a leaf are all that will be required by the mobile application to identify common diseases such as Early Blight, Late Blight within seconds to enable farmers to act on time before the disease spreads. Not only does this help the crop as a whole, but overall agricultural output and earnings will increase. Additionally, it decreases dependence on resource from outside hence; it enhances self reliance and confidence in decision making by farmers.

The system is especially helpful to smallholder farmers who typically do not have access to modern diagnostic equipment or agricultural extension. It minimizes the knowledge gap while making digital technology more inclusive as it delivers AI powered solutions in a form that is portable, accessible and easy to use. In this manner, the application has potential to improve both productivity and quality of life in rural farming territories by fostering resilience, minimizing risk and fostering sustainable livelihoods.

5.2.2 Impact on Society & Environment

On a broader scale, the system supports the goals of digital agriculture and precision farming. It promotes the use of data-driven decision-making and automation in agricultural practices, helping to modernize and professionalize the farming sector. By integrating advanced computer vision models like YOLOv8 and YOLOv9 into a mobile application, the system demonstrates how artificial intelligence can be brought to the hands of ordinary farmers, facilitating a more equitable distribution of technological benefits.

Environmentally, the application encourages the use of pesticides because it prevents their excess use and misuse. In the standard settings, farmers usually tend to use preventive spraying because there is little accurate information on when disease occurs. It gives rise to unwarranted chemical application, which may harm useful microbes, pollute bodies of waters through run off, and cause destruction of soils over time. With early and precise diagnosis of the disease then farmers can apply the treatments when and where needed in appropriate doses. This specific intervention becomes a catastrophe preventative and minimizes the environmental impact of potato cultivation.

Besides, the lightweight use of a TFLite model for mobile implementation provides energy effectiveness. Compared to legacy systems that rely on vast computing powers or constant internet connectivity this solution does all processing locally on the device so it consumes much less energy and carbon footprint. This design decision is especially relevant to developing green technology and lowering the ecological costs of digital agriculture.

5.2.3 Ethical Aspects

The ethical basis of this project is responsible AI development, and responsible use of data. All images (both usage and testing) were obtained from publicly available datasets or were obtained under proper guidelines without breaking any copyright or privacy. No personal information of sensitive users was stored or shared through the course of model development or app deployment.

Furthermore, the system design promises the privacy of data and control by the

user. Because of the disease detection that occurs in a non-online manner using the TFLite model on the users device, there is no image uploaded to a remote server thus eliminating risks such as breach of data or unauthorized use. This is particularly true for rural and under represented communities, and digital trust must be cultivated in such communities to promote radicalization of new technology. The app interface is intentionally designed to be simple, language-independent (through visual cues), and easy to navigate, making it usable by individuals with limited digital literacy.

5.2.4 Sustainability Plan

Sustainability is a core component of the project design, both in terms of environmental conservation and long-term technical scalability. By using open-source platforms like Flutter, Android Studio, and Google Colab, the system is developed in a cost-effective and maintainable manner. These tools not only reduce the economic barrier to innovation but also allow future developers to extend the system with minimal additional resources. The modular structure of the application supports scalability. As new datasets or disease classes become available, the model can be retrained and updated without needing to rebuild the app from scratch. This makes the system future-ready and adaptable to emerging agricultural challenges such as new diseases, pest outbreaks, or application to other crops beyond potatoes.

In addition, the ability of the app to operate offline avoids intensive use of cloud infrastructure or high bandwidth networks. This reduces repeated costs and makes the system viable in remote areas with poor connections. Looking at the angle of energy consumption, the lightweight design also means minimal mobile resource consumption, hence saving the battery and sustaining longer field use. In essence, this project shows how AI empowered agricultural solutions may be built responsibly and sustainably so that they are not just technically sound, but also socially equitable and environmentally conscious.

5.3 Project Management and Financial Analysis

Optimal project functioning involves good planning, implementation of tasks, time sharing, and cost appraisal. This section presents the method of project

organization applied in the development of the proposed system, the task scheduling, the development stages, and the use of resources and cost elements. As the research combines deep learning, a mobile-based deployment strategy, and the ability to manage computational, human, and economic resources in ways that ensured optimal use of resources for time and budget constraints was critical to achieving the required outcomes in the time and budget frame.

5.3.1 Project Planning and Task Management

Successful planning, development and deployment of the YOLO-based potato leaf disease detection system had effective project management as a key element. The project has applied phased development, dividing the entire work into several clearly delineated stages: problem identification, through review of literature, collection and annotation of dataset, model training and evaluation, development of mobile app, an eventual deployment.

Each phase had clearly assigned deliverables, milestones, and deadlines. Initially, the research problem was explored through a literature review to identify the limitations of existing approaches. This guided the selection of YOLOv8, YOLOv9, and YOLOv10 for model experimentation. The dataset was collected and annotated during the early development phase, and training was conducted using Google Colab's GPU support. Parallely, the mobile application interface was prototyped using Flutter, ensuring synchronization between model output and app design.

To manage progress, a task allocation plan was developed, distributing responsibilities across data preparation, model training, testing, conversion to TFLite, app integration, and documentation. Agile methods were loosely followed—progress was reviewed weekly, and changes in the training pipeline or app features were introduced iteratively based on performance analysis.

Time management and version control tools like Google Drive and GitHub were used to store, monitor, and back up code, model checkpoints, and documentation files.

5.3.2 Tools and Platforms Used

Table 5.1 provides an overview of the essential tools and platforms used for model

training, development, deployment, and version control throughout the project.

Table 5.1: Details of tools and platform used

Category	Tool/Platform	Purpose
Model Training	Google Colab (GPU)	Training models
Programming Language	Python	Core DL development and preprocessing
DL Framework	TensorFlow / Keras	Model building, training, and TFLite conversion
Mobile Development	Flutter	Cross-platform mobile app development
IDE	Android Studio	Testing and packaging the Android app
Version Control	Git / GitHub	Code versioning and collaboration

5.3.3 Financial Analysis

This project was designed to be cost-effective and accessible for students or independent researchers. Below is an approximate cost breakdown:

Table 5.2: Estimated Cost and Financial Analysis.

Resource/Item	Estimated Cost (BDT)	Remarks
Google Colab Pro (optional)	3500	Optional for extended GPU time and faster training
Computer (personal or lab use)	Existing Resource	Used for app development and testing
Internet Access	4200	Required for cloud training and resource access
Flutter and Android Studio (free tools)	0	Open-source and community-supported development tools
Cloud Storage/Backup (optional)	300	Google Drive or similar platforms
Total	8000 BDT	

5.4 Complex Engineering Problem

This section highlights the complex nature of the engineering problem addressed in this project. The task of designing a highly accurate, efficient, and mobile-compatible potato leaf disease detection system involves integrating multiple areas of knowledge, managing technical constraints, and ensuring smooth deployment in real agricultural environments. It requires a combination of deep learning model training, mobile app

development using Flutter, real-time inference via TensorFlow Lite, and adherence to performance, ethical, and deployment standards. The following subsections provide detailed mappings to problem-solving categories, knowledge profiles, and engineering activities, showcasing the depth and interdisciplinary nature of the work.

5.4.1 Complex Problem Solving

To successfully complete this project, several complex problem-solving aspects had to be addressed—from selecting the right model architecture to real-time mobile deployment. The table below maps the engineering problem with standard complexity attributes:

Table 5.3: Mapping with Complex Problem Solving

EP1	EP2	EP3	EP4	EP5	EP6	EP7
✓		✓	✓			✓

Justifications:

- **EP1 – Depth of Knowledge:** The project required strong knowledge of deep learning (YOLOv8/v9/v10), image annotation, TFLite model conversion, and Flutter-based app development. Skills in Python programming, transfer learning, and mobile system integration were essential.
- **EP3 – Depth of Analysis:** A comparative analysis of different YOLO models was conducted. Accuracy, model size, inference time, and mobile compatibility were all evaluated to select the optimal model for deployment.
- **EP4 – Familiarity of Issues:** Challenges such as class imbalance, model optimization, and mobile integration were well understood through prior academic exposure and personal projects, aiding effective problem-solving.
- **EP7 – Interdependence:** The solution is modular, allowing the system to be expanded to other crops or plant diseases, making the framework scalable and interdependent with broader agricultural AI applications.

Mapping with Knowledge Profile for EP1

Table 5.4: Mapping with Knowledge Profile

K3	K4	K5	K6	K8
✓		✓		✓

Justifications:

- K3 – Engineering Fundamentals: The project involved mathematical concepts from machine learning, computer vision, and statistics for training and validating the YOLO-based detection model.
- K5 – Engineering Design: From system architecture planning to designing the UI/UX of the Flutter mobile app, this project involved end-to-end engineering design customized for real-world agricultural use.
- K8 – Research Literature: Existing research papers and benchmarking studies played a major role in selecting appropriate models, setting training parameters, and identifying performance metrics.

5.4.2 Engineering Activities

The system required a series of engineering activities including data preprocessing, model training, mobile deployment, performance tuning, and app-user integration. The mapping below reflects how the project relates to complex engineering activity indicators:

Table 5.5: Mapping with Complex Engineering Activities

EA1	EA2	EA3	EA4	EA5
✓		✓	✓	✓

Justifications:

- EA1 –The project utilized a variety of tools—Google Colab for model training, TensorFlow for inference, Flutter for mobile app development, and Android Studio for testing and debugging.

- EA3 –The integration of YOLOv10 with real-time disease detection and mobile compatibility is a novel contribution in the context of potato farming in rural areas.
- EA4 –The project aims to reduce pesticide misuse, improve crop yield, and empower farmers through accessible AI technology, contributing to social and environmental sustainability.
- EA5 – Skills developed through academic projects, deep learning coursework, and hands-on app development experience directly supported the completion of this work.

5.5 Summary

This chapter has covered the engineering standards, design challenges and complexity of a real-time, mobile compatible potato leaf disease detection system. The project required many levels of technical implementation including deep learning model training via YOLO variants, image annotation, development of mobile application using Flutter, and lightweight model deployment through TensorFlow Lite. Conformity to the standards of software, hardware and communication guarantees that the solution was robust, efficient, and applicable for practical agricultural use. With the use of open-source tools including; Python, Google Colab, Android Studio and Flutter for developing a cheaper and scalable system. The model was tailored to the use on Android devices and allowed offline inference without trashing user privacy and performance reliability.

The wider societal, environmental and ethical implications of this system were analyzed too. The project aims at reducing pesticide overuse, equipping farmers with timely detection of diseases and development of digital agriculture in poor communities. Ethical design-routine measures such as local model inference and minimum user data access were preferred during development.

Moreover, the chapter emphasized the interdisciplinary nature of the project by correlating the project with complex engineering problem solving attributes, knowledge profiles, and engineering activities. These mappings illustrated how the work fuses high-end machine learning, mobile app development, and responsible deployment practices into solving a real-world agricultural issue.

Chapter 6

Conclusion

This chapter ends the research by summarizing the main findings of the research and the contributions made through the development. It also outlines the challenge encountered during implementation and provides some directions to which future activities could be directed. The chapter answers the question on the practical impact of the system and how it can be advanced to form a more complete, interpretable and scalable solution for real-time medical diagnosis with available mobile technology.

6.1 Summary

This thesis investigated the development of a real time, deep learning based system to detect potato leaf diseases using state-of-the-art object detection models and mobile application deployment. The work was motivated by practical challenges farmers encounter; in detecting and controlling crop diseases, particularly in low-resource situations wherein expert help and diagnostic tools may be lacking. In an effort to fill this gap, the study suggested a full pipeline that aggregates YOLO-based deep learning models, image annotation, model evaluation and mobile app development using Flutter.

The project started with the construction of a well-annotated set of potato leaf images with disease cases like Early Blight and Late Blight and healthy samples. Image annotation was done to make object detection based training possible. Three state-of-the-art YOLO models YOLOv8m, YOLOv9m, and YOLOv10m were trained and tested, in order to identify the best performing model with regards to accuracy and speed, for real time detection. After thorough analysis, the highest performing model was chosen and converted to a format suitable for mobile deployment which is TensorFlow Lite (.tflite). To bring the solution into practical use, a cross-platform Flutter-based mobile application was developed. This app allows users to capture or upload an image of a potato leaf and instantly receive disease detection results, with bounding boxes visually identifying

affected areas. The study also explored the broader engineering aspects of the project, including software and hardware compliance, ethical considerations, environmental impact, and the interdisciplinary complexity involved in integrating AI with real-world agriculture. The engineering problem was mapped to standard complexity metrics, demonstrating the depth and innovative nature of the solution.

6.2 Limitation

While the system achieved promising results and practical deployment, several limitations were observed during the course of development:

- **Limited disease classes:** The current system is trained only on a few classes (Early Blight, Late Blight, Healthy). Other important potato diseases such as Leaf Roll or Black Leg are not included due to lack of labeled data.
- **Dataset size and diversity:** The training dataset, although manually annotated, is relatively limited in size and lacks diversity in terms of environmental conditions, lighting variations, and background noise, which may impact real-world accuracy.
- **Model interpretability:** Although YOLO is highly effective for detection, the model operates as a black box. Explainable AI tools like LIME or SHAP were not integrated, which may reduce user trust or understanding of predictions.
- **Device constraints:** Although the application runs on most mid-range Android smartphones, older or low-memory devices may experience performance issues or slow inference.
- **No multilingual support:** The app interface is currently in English only, which may limit its usability among non-English-speaking farmers.

These limitations serve as important considerations for improving future iterations of the system and expanding its practical relevance.

6.3 Future Work

Several directions exist for development of this work and adding more functionality:

- Expansion of disease classes: The potato leaf diseases dataset can be expanded to others such as Potato Leaf Roll Virus (PLRV), Verticillium Wilt and Bacterial Wilt. This will enhance the diagnostic power of the model and give the model more comprehensiveness.
- Cross-crop model generalization: Slight modifications in dataset and retraining, can then make the same framework applicable for other crops such as tomato/chilli/mango with again use of same YOLO detection strategy.
- Integration of explainable AI (XAI): [Tools such as Grad-CAM, LIME, or SHAP can be integrated into the mobile application to provide visual or textual explanation of predictions thus increasing transparency and trust to the people using it.Ça a.*l]
- Field-testing and user feedback: Real world field trials with farmers and agricultural experts can test the system under the natural conditions. Impulse response and responses in other environments can be guided by user feedback.
- Multilingual and voice-based support: Providing both local language options and voice prompts will increase the access ability, particularly for the elderly or low-literacy farmers.
- Updates by cloud; remote model tuning: Although the app works offline the future versions will allow for optional cloud connections that will allow for periodic model updates/data syncing for research and extension services.
- Integration with the Internet of Things and sensor system: For a complete smart farming solution the app can be linked to soil sensors, weather data and irrigation systems to see the overall picture of the health of the crop and farming in general.

With the building blocks provided by this work, the system can further grow into a solid agricultural assistant, to provide sustainable, technology-grounded farming on a factory scale.

References

- [1] Sholihati, R. A., Sulistijono, I. A., Risnumawan, A., & Kusumawati, E. (2020, September). Potato leaf disease classification using deep learning approach. In 2020 international electronics symposium (IES) (pp. 392-397). IEEE.
- [2] Kumar, A., & Patel, V. K. (2023). Classification and identification of disease in potato leaf using hierarchical based deep learning convolutional neural network. *Multimedia Tools and Applications*, 82(20), 31101-31127.
- [3] Rahman, M. M., Yeasmin, T., Islam, M. J., Mahmud, M. D., & Rahman, M. M. (2023). Potato Leaf Disease Prediction: A Machine Learning Perspective. *Journal of Scientific and Technological Research*, 5(1), 27-35.
- [4] Rashid, J., Khan, I., Ali, G., Almotiri, S. H., AlGhamdi, M. A., & Masood, K. (2021). Multi-level deep learning model for potato leaf disease recognition. *Electronics*, 10(17), 2064.
- [5] Paul, H., Ghatak, S., Chakraborty, S., Pandey, S. K., Dey, L., Show, D., & Maity, S. (2024). A study and comparison of deep learning based potato leaf disease detection and classification techniques using explainable AI. *Multimedia Tools and Applications*, 83(14), 42485-42518.
- [6] Shiffa, T. S., & Suchithra, M. S. (2023, December). Potato Leaf Diseases Detection Using Machine Learning And Deep Learning. In 2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS) (pp. 1-6). IEEE.
- [7] Gupta, U., Vijh, S., Kumar, S., Roy, N. R., & Jodan, J. S. (2023, November). Potato Leaf Disease Detection Using Machine Learning Techniques for Precision Agriculture. In 2023 Seventh International Conference on Image Information Processing (ICIIP) (pp. 913-918). IEEE.
- [8] Tambe, U. Y., Shobanadevi, A., Shanthini, A., & Hsu, H. C. (2023). Potato Leaf Disease Classification using Deep Learning: A Convolutional Neural Network Approach. arXiv preprint arXiv:2311.02338.
- [9] Tarik, M. I., Akter, S., Al Mamun, A., & Sattar, A. (2021, February). Potato disease detection using machine learning. In 2021 Third international conference on intelligent communication technologies and virtual mobile networks (ICICV) (pp. 800-803). IEEE.
- [10] Islam, M. A., & Sikder, M. H. (2022). A deep learning approach to classify the potato leaf disease. *Journal of Advances in Mathematics and Computer Science*, 37, 143-155.

- [11] Sharma, S., Anand, V., & Singh, S. (2021, June). Classification of diseased potato leaves using machine learning. In 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT) (pp. 554-559). IEEE.
- [12] Nishad, M. A. R., Mitu, M. A., & Jahan, N. (2022). Predicting and classifying potato leaf disease using k-means segmentation techniques and deep learning networks. *Procedia Computer Science*, 212, 220-229.
- [13] Yu, T., Zhou, J., Fan, J., Wang, Y., & Zhang, Z. (2023). Potato leaf area index estimation using multi-sensor unmanned aerial vehicle (UAV) imagery and machine learning. *Remote Sensing*, 15(16), 4108.
- [14] Kadam, S. U., Dhede, V. M., Khan, V. N., Raj, A., & Takale, D. G. (2022). Machine learning method for automatic potato disease detection. *NeuroQuantology*, 20(16), 2102-2106.
- [15] Iqbal, M. A., & Talukder, K. H. (2020, August). Detection of potato disease using image segmentation and machine learning. In 2020 international conference on wireless communications signal processing and networking (WiSPNET) (pp. 43-47). IEEE.
- [16] Singh, M., Bhatt, S., Chaudhary, P., & Verma, A. (2024, July). Disease Identification in Potato Leaf using Deep Learning. In 2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC) (pp. 911-916). IEEE.
- [17] Mahum, R., Munir, H., Mughal, Z. U. N., Awais, M., Sher Khan, F., Saqlain, M., ... & Tili, I. (2023). A novel framework for potato leaf disease detection using an efficient deep learning model. *Human and Ecological Risk Assessment: An International Journal*, 29(2), 303-326.
- [18] Khalifa, N. E. M., Taha, M. H. N., Abou El-Maged, L. M., & Hassanien, A. E. (2021). Artificial intelligence in potato leaf disease classification: a deep learning approach. *Machine learning and big data analytics paradigms: analysis, applications and challenges*, 63-79.
- [19] Radwan, M., Alhussan, A. A., Ibrahim, A., & Tawfeek, S. M. (2024). Potato Leaf Disease Classification Using Optimized Machine Learning Models and Feature Selection Techniques. *Potato Research*, 1-25.
- [20] Sharma, S., Anand, V., & Singh, S. (2021, June). Classification of diseased potato leaves using machine learning. In 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT) (pp. 554-559). IEEE.
- [21] Basak, S. K. (2023). Leaf Disease Detection (Potato Leaf) Using Machine Learning (Doctoral dissertation, University of Engineering & Technology).

211-15-4077

ORIGINALITY REPORT

17%

SIMILARITY INDEX

10%

INTERNET SOURCES

11%

PUBLICATIONS

8%

STUDENT PAPERS

PRIMARY SOURCES

- 1 Submitted to Daffodil International University 3%
Student Paper
- 2 dspace.daffodilvarsity.edu.bd:8080 2%
Internet Source
- 3 www.mdpi.com 1%
Internet Source
- 4 d197for5662m48.cloudfront.net 1%
Internet Source
- 5 Prit Mhala, Anushka Bilandani, Sanjeev Sharma. "Enhancing crop productivity with fined-tuned deep convolution neural network for Potato leaf disease detection", Expert Systems with Applications, 2024 <1%
Publication
- 6 "Advances in Data-Driven Computing and Intelligent Systems", Springer Science and Business Media LLC, 2024 <1%
Publication
- 7 Submitted to United International University <1%
Student Paper
- 8 www.coursehero.com <1%
Internet Source
- 9 "Advances in Artificial-Business Analytics and Quantum Machine Learning", Springer Science and Business Media LLC, 2024 <1%
Publication