Advancing Image Classification through Self-Supervised Representation: Comparative Benchmarking with Lightweight CNNs, and Traditional Baselines

By

Supta Das Dip
213-15-4598

Mohammad Asif Dewan
213-15-4548

FINAL YEAR DESIGN PROJECT REPORT

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

Supervised by

Md Sazzadur Ahamed
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

Shah Md. Tanvir Siddiquee
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

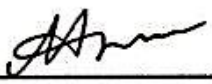
September 17, 2025

APPROVAL

This Project titled **"Advancing Image Classification through Self-Supervised Representation: Comparative Benchmarking with Lightweight CNNs, and Traditional Baselines"**, submitted by **Supta Das Dip** and **Mohammad Asif Dewan**, ID No: **213-15-4598** and **213-15-4548** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **17 September, 2025**.

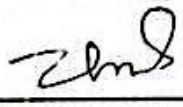
BOARD OF EXAMINERS

Dr. S.M Aminul Haque — **Chairman**
Professor and Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Ms. Nazmun Nessa Moon — **Internal Examiner**
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Md. Abbas Ali Khan — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

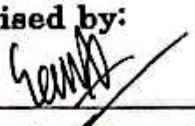


Dr. Md. Zulfiker Mahmud — **External Examiner**
Professor
Department of Computer Science and Engineering
Jagannath University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Md Sazzadur Ahamed, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this projectnor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



Md Sazzadur Ahamed
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Shah Md. Tanvir Siddique
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Submitted by:

Supta Das Dip

Supta Das Dip
Student ID: 213-15-4598
Department of Computer Science and Engineering
Daffodil International University

Mohammad Asif Dewan

Mohammad Asif Dewan
Student ID: 213-15-4548
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 **Md Sazzadur Ahamed**, **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 **Computer Vision** 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

Image-based plant disease classification has become increasingly important in supporting maintainable agriculture by enabling early detection and intervention. However, existing approaches often face two critical limitations: dependency on heavy data augmentation with the issue caused by class imbalance, when working dataset is limited. This proposed work addresses these challenges through a dual-pipeline framework applied to bean and bean leaf disease classification. The influences self-supervised learning using SimCLR with lightweight CNN backbones (ResNet18 and DenseNet121) to learn robust feature representations directly from raw image distributions, minimizing dependency on large-scale labeling and augmentation at the begining. On the other hand, Sobel-Feldman edge detection and contour analysis were applied through classical image processing with custom-made features contribute to machine learning classifiers such as XGBoost to provide benchmarking baselines model. The dataset contains eight classes, with significant imbalance between majority and minority classes. The experimental results demonstrate that the SimCLR-based pipeline achieves strong generalization for underrepresented classes also and outperforming traditional ML baselines in terms of performance like accuracy, macro f1-score, and precision-recall balance. Here we can say, the classical pipeline contributes clarity and lightweight deployment potential. The experimental findings demonstrates that the effectiveness of self-supervised learning in reducing augmentation dependency, reducing imbalance issue, and providing impartial classification over classes. This work not only improves methodological insights but also delivers a practical and scalable solution for bean disease detection, contributing to future agricultural research and real-world farming applications.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image classification has become important in agricultural and plant disease research that monitoring of plant health by fast detection. In this chapter, we will introduce about the development of deep learning and computer vision techniques, automatic recognition of diseases in crops and their leaves which has achieved major attention. This study plans to address the challenges like class imbalance, dependency on data augmentation by applying self-supervised learning. Also benchmarking comparison against traditional image processing and machine learning approaches.

1.1 Introduction

Plant and crop disease identification contains detecting and investigating infections that affects agricultural productivity. Identification of plant and crop disease is essential agricultural growth and effective yield management. Traditional inspection is time-consuming, likely to human error, and infeasible for large-scale monitoring. Automated image-based classification provides a positive alternative by using computational methods to detect small differences in disease patterns. This process should not be dependent on the image size, numbers, or balanced label which may causes for data memorizing and unpredictable for unseen data. However, well known technique like data augmentation is a process that can minimize the data imbalance issue but it reduces the image quality also. While conventional supervised learning methods heavily rely on large labeled datasets and extensive data augmentation, which may not always be available or feasible, especially for rare diseases or minor classes [1].

Class imbalance in datasets can significantly hinder model performance, causing biased predictions toward majority classes. In that case, self-supervised learning techniques have emerged as a promising solution. The limitation that mentioned earlier, motivates the exploration of self-supervised learning methods, such as SimCLR (Simple Framework for Contrastive Learning of Visual Representations), which can learn robust feature representations with reduced dependency on labeled data and augmented images. Contrastive learning is applied to make different augmented views of the same image similar while keeping them distinct from other images. Thus, SimCLR does not require explicit class labels during pretraining, allowing it to learn generalizable features directly from the raw image distribution [2].

The core concept of Simple Framework for Contrastive Learning of Visual

Representations (SimCLR) is to encode input images into a latent space using a convolutional or transformer model, followed by a projection head. With a contrastive loss function, the model brings representations of the same image closer together with the positive pairs while pushing those from different images further apart from the negative pairs. The resulting feature space of this process is highly discriminative and robust, even before fine-tuning on labeled data [3].

A major advantage of using SimCLR in the bean and bean leaf dataset is its ability to alleviate class imbalance issues without relying on oversampling techniques like SMOTE or heavy artificial data augmentation. Since the model learns from the natural structure of the data rather than directly from the class distribution and minority class samples which contributes equally to representation learning during pretraining. The data distribution of bean and bean leaf data is quite imbalanced and the margin of imbalance ratio is 19.4 between the higher and lower classes. So, the self-supervised technique also works with the classes with limited samples, which benefits from strong feature extraction and reducing the biasness toward majority classes. Fine-tuning with a smaller set of labeled examples, SimCLR provides a balanced representation that improves classification decently through every classes [4].

Combining these modern approaches with classical feature extraction techniques such as Sobel-Feldman edge detection allows for comparative benchmarking, pointing out the practical strengths and limitations of each method.

1.2 Motivation

There are more studies on plant disease classification, most of them focuses on standard supervised learning or popular CNN architectures which often producing similar outcomes with little novelty. Most of the work has been done on huge computational power and well config device that produce higher performance rate with transformer based or heavy weight models. The currency of class imbalance in datasets, as seen in bean and bean leaf images, present a major challenge for accurate classification. Most of the work rely strongly on data augmentation to mitigate this issue, but the augmentation technique can introduce low-image quality or fail to capture real-world variations [5].

This study is motivated by the need for a more generalizable, data-efficient, and comparative approach that leverages self-supervised learning to reduce augmentation dependency while still ensuring decent classification performance with low-end devices and lightweight model application. Additionally, integrating classical ML pipelines as comparative benchmarking which provides a deeper understanding of the strengths and weaknesses of modern methods in practical scenarios.

1.3 Objectives

The primary objectives of this study are:

i. To develop a self-supervised representation learning pipeline using SimCLR for bean and bean leaf image classification, minimizing the dependency on heavy data augmentation.

ii. To address class imbalance issues in the dataset and assess how self-supervised learning mitigates this challenge.

iii. To perform comparative benchmarking with lightweight CNNs using SimCLR approach and traditional machine learning models using Sobel-Feldman feature extraction.

1.4 Methodology

The project adopts a dual-pipeline strategy to address the challenges of bean and bean leaf disease classification, particularly the issues of class imbalance and dependency on heavy data augmentation. Additionally, the project will monitor the computation power used and comparative benchmarking of the device config usage.

The first pipeline focuses on self-supervised learning using Simple Framework for Contrastive Learning of Visual Representations, a modern approach that allows models to learn meaningful feature representations directly from raw image data without requiring extensive labeling or data augmentation. Since SimCLR learns from the natural structure of the dataset rather than being constrained by class labels during pretraining, minority class samples contribute equally to the learned representations. As a result, underrepresented classes benefit from the same level of feature learning as majority classes.

The second pipeline establishes a classical feature-based benchmarking approach. Here, Sobel-Feldman edge detection is applied to extract meaningful structural and texture-related features from the images. These handcrafted features are then fed into lightweight machine learning classifiers, serving as traditional baselines against which the performance of the SimCLR-based self-supervised pipeline can be compared.

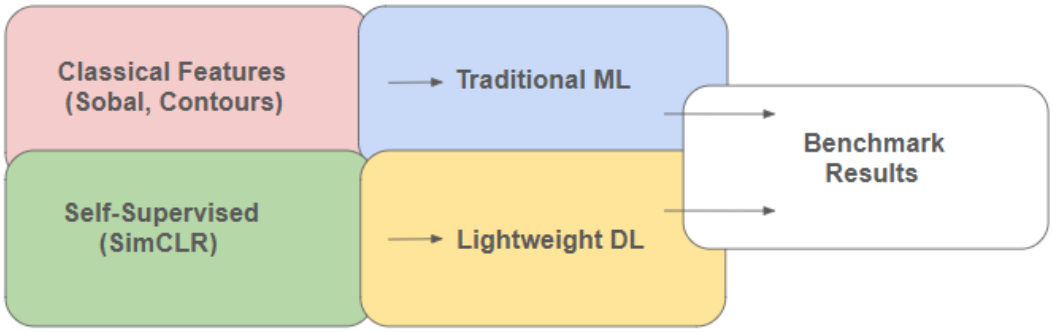


Figure 1.1: Methodology Pipeline Diagram

Figure 1.1 describes both pipelines are trained and evaluated on the same dataset, which comprises eight classes divided into two categories: six classes of bean leaves and two classes of beans. This dual-pipeline approach not only demonstrates the effectiveness of SimCLR in reducing augmentation dependency and handling imbalance but also provides meaningful benchmarking against classical methods.

1.5 Project Outcome

The expected outcomes of this research include:

- A robust self-supervised learning model capable of classifying bean and bean leaf images accurately without any augmentation process.
- An enhanced understanding of class imbalance handling in image classification tasks.
- Provide a comparative analysis between modern self-supervised methods and traditional feature-based ML approaches.
- Contribution of practical knowledge and guidelines for future agricultural image-based disease detection research.

1.6 Organization of the Report

This report is organized into six comprehensive chapters, each focusing on a specific aspect of the research process and findings.

i. **Chapter 1 Introduction:** Presents the background of the study, outlines the motivation for addressing class imbalance and dependency on augmentation, defines the research objectives, introduces the dual-pipeline methodology, and highlights the expected contributions of the project.

ii. **Chapter 2 Background:** Reviews the relevant literature on plant disease detection, image classification methods, self-supervised learning approaches, and traditional image processing techniques such as Sobel-Feldman. This chapter also identifies research gaps that justify the adoption of SimCLR for this study.

iii. **Chapter 3 Research Methodology:** Provides a detailed description of the adopted dual-pipeline framework. It explains dataset preparation, preprocessing steps, the design of the SimCLR-based self-supervised pipeline, the implementation of the classical Sobel-Feldman feature extraction with ML classifiers, and the evaluation strategies used.

iv. **Chapter 4 Implementation and Results:** Describes the experimental setup, training procedures, and implementation details for both pipelines. This chapter presents the obtained results, including performance metrics such as accuracy, F1-score, and precision-recall, followed by comparative analysis between self-supervised learning, CNNs, and traditional baselines.

v. **Chapter 5 Engineering Standards and Design Challenges:** Discusses the technical standards and best practices followed in system implementation, as well as ethical considerations in dataset usage, the societal impact of agricultural disease detection, and challenges such as class imbalance, computational constraints, and scalability.

vi. **Chapter 6 Conclusion:** Summarizes the key contributions and findings of the research, reflects on limitations faced during experimentation, and suggests future directions such as exploring other self-supervised frameworks, expanding the dataset, or applying the approach to different crops and agricultural domains.

Chapter 2

Background

Image classification has become a fundamental technique in agricultural research, particularly for detecting and monitoring plant health. In recent years, the integration of computer vision and machine learning has enabled automated diagnosis of plant diseases from image data, offering a scalable solution to support farmers and researchers. This chapter provides a complete background on image-based disease detection, focusing on methods applied to beans and bean leaves. It discusses the major challenges in agricultural image classification, such as dataset imbalance, dependency on augmentation, and limited generalization of existing models. The chapter reviews the prior work in deep learning and classical machine learning for plant disease classification, finding the gaps that motivate the use of self-supervised approaches like SimCLR. Finally, it outlines how this research contributes by combining self-supervised representation learning with traditional baselines for comparative benchmarking

2.1 Introduction

Plant diseases create an important threat to global food security, reducing crop yield and affecting both smallholder and large-scale farmers. Early and accurate detection of such diseases is essential for effective management and long-term agricultural practices. Traditional methods of disease detection rely on human inspection, which is not only time-consuming but also likely to human error. Image-based classification techniques have emerged as a practical alternative, introducing advances in computer vision and machine learning to identify small disease patterns in plant leaves and crops.

A major limitation in the field is the lack of methods that generalize well across different data distributions. Most plant disease classification models are trained in a fully supervised approach that requires large labeled datasets that are often expensive and difficult to collect, especially in agricultural contexts. This dependency on labels restricts the applicability of supervised CNN-based approaches in low-data scenarios.

Self-supervised learning (SSL) has recently gained attention as a powerful example by addressing these gaps for representation learning without heavy dependency on annotated data. SimCLR (Simple Framework for Contrastive Learning of Visual Representations) has demonstrated amazing effectiveness in learning transferable and unfair features from unlabeled images among these methods. SimCLR provides a promising route by reducing dependency on labels and data augmentation to

overcome both class imbalance and generalization issues in agricultural image classification.

Classical feature extraction techniques such as Sobel-Feldman edge detection and contour detection remain valuable for benchmarking, as they offer understanding, computationally efficient baselines. Integrating both modern and traditional approaches allows this work to evaluate the strengths and limitations of each method completely.

This section sets the foundation for the study by reviewing existing methods in plant disease classification, identifying their limitations, and establishing the need for a dual-pipeline approach that combines self-supervised representation learning with comparative benchmarking against lightweight CNNs and traditional ML classifiers.

2.2 Literature Review

A reasonable amount of research has explored the use of convolutional neural networks (CNNs) for plant disease detection with architectures such as VGG, ResNet, and DenseNet achieving promising results. Traditional machine learning methods based on manual features such as edge detection, color histograms, and texture descriptors that have also been applied though with limited scalability compared to deep learning. Despite these developments, existing approaches often suffer from two recurring challenges, class imbalance within datasets and heavy reliance on data augmentation to artificially increase sample variety. When certain diseases have very few samples, models tend to overfit to dominant classes which leading to biased predictions. Augmentation-based strategies may introduce nonnatural variations that do not always reflect real-world conditions at the same time.

A. Katumbo et al. [6] presented a real-time plant disease detection framework that integrates edge computing using deep learning, mainly identifying bean plant pathologies. To support that, the author collected a bean plant datasheet and a pathologies datasheet containing annotated leaf images obtained under natural field conditions. The approaches involve training convolutional neural networks, including lightweight variants of ResNet and MobileNet, which were developed on Raspberry Pi-based edge devices. This architecture focused on reducing dependency on cloud resources by ensuring low latency, which helps smallholder farmers in low-resource environments. The author examined performance in relation to interface speed and found that MobileNet-based models offered the best balance for real-time use. The system was found to be scalable and able to connect with IoT farming systems. This study offers an affordable and simple solution for plant disease detection, also controlling and showcases how edge AI can support advancing

agricultural farming technologies.

M. V. Shewale et al. [7] focused on designing a deep learning architecture ready made for early detection and classification of plant diseases. The proposed model used a fine-tuned convolutional neural network (CNN) with surviving connections to improve feature extraction from high-resolution leaf images. The Plant Village dataset was applied and tested, which contains multiple crop species and disease categories. The author applied image augmentation methods to avoid overfitting so that the model can work well under different field conditions. The model outperformed baseline CNNs and transfer learning models, achieving classification accuracy exceeding 99% accuracy on test data. The study highlighted the importance of early detection for preventing yield loss. At the same time, the authors also admitted the datasheet was limited, as it didn't represent real-life farming conditions, such as poor lighting and leaf overlap. Future research was suggested to include more realistic datasets. Overall, this model offers a high accuracy and practical solution for plant disease detection.

K. M. Hosny et al. [8] proposed a multi-class classification method for plant leaf disease detection through a hybrid model that combines both ensemble and deep features. The dataset included about 3,171 to 3,948 leaf images from different plant species and disease types, which were later divided into training, validation, and test sets. To capture both overall and detailed information of the diseased leaves, the authors combined Local Binary Pattern (LBP) texture features with CNN-based features. This combined representation improved the model's performance compared to CNN alone. The experiments showed that the hybrid model achieved very high accuracy, between 99% and 99.6%. The confusion matrix results also showed that this method worked well for diseases with similar appearances, where a single CNN model often made mistakes. The best-performing model was the CNN+LBP hybrid model, which provided both accuracy and Clearer results. However, the study was limited because it used only a single dataset and did not test on field data, which makes generalization uncertain. In conclusion, the study showed that combining features is an effective way to improve deep learning models for detailed recognition of plant leaf diseases.

E. Moupojou et al. [9] introduced FieldPlant, a new dataset created to make plant disease detection models more realistic and useful for real-world farming. The dataset contained 5,170 field images collected from cassava, corn, and tomato plants, with expert annotation identifying 8,629 diseased leaves across 27 disease types. The data was carefully checked and organized using the RoboFlow platform. The authors trained several deep learning models such as MobileNet, EfficientNet, and YOLO for testing the dataset. The results after testing showed that while models achieved over 95% accuracy on controlled datasets, their performance dropped to 65–77% on the FieldPlant dataset. This drop was mainly due to real-life problems like background clutter, poor lighting,

and overlapping leaves. Models trained with stronger augmentation performed little bit better, but still could not reach laboratory-level results. The best-performing models were lightweight architectures with heavy augmentation but they still struggled with background noise. A critical factor for field-ready detection that the authors emphasized that data realism rather than only focusing on model architecture. The main limitation of the study was that it included only three crops, leaving room for expansion. FieldPlant is an important resource for developing robust and scalable plant disease detection systems.

M. O. Ojo and A. Zahid [10] developed a plant disease detection system focusing on improving accuracy by using preprocessing methods and handling data imbalance. The dataset contained tomato leaf images collected under greenhouse conditions, which were divided into balanced and imbalanced sets for testing. Different preprocessing techniques such as Adaptive Histogram Equalization (AHE), Contrast Limited AHE (CLAHE), image sharpening, and a combination of CLAHE with improving were used by the authors. Oversampling techniques such as SMOTE, major-to-minor translation (M2m), and GAN-based augmentation were also employed. Deep learning models like InceptionV3, Xception, ResNet101, and ResNet50 were trained and evaluated. The results showed that the best outcome came from combining CLAHE with sharpening and GAN augmentation, where ResNet-50 achieved 97.69% accuracy and an F1-score of 97.62%. GAN augmentation performed better than SMOTE or M2m after every testing. The study was limited because it only tested on greenhouse images and not on real field data. It was shown that sometimes more than changing the model, proper preprocessing and balancing methods can improve plant disease detection.

M. Alsaidi et al. [11] addressed the pervasive issue of class imbalance in image classification using a dermoscopic dataset, with lessons directly applicable to agricultural disease detection. The HAM10000 dataset contained 10,015 images distributed across seven classes, heavily concentrated in a single dominant category. The authors used an 80:20 split for training and testing. For balancing, 36,981 augmented images and 36,926 GAN-generated samples were added to the minority classes. Transfer learning architectures such as VGG16, VGG19, ResNet, and Inception were trained under both augmentation and GAN regimes. After testing, the results showed that augmentation produced more stable gains than GAN, with the best performance achieving 93.2% accuracy using ResNeXt101. GAN-generated samples, while visually realistic, did not achieve the same improvements on minority classes as traditional augmentation. The best-performing model was ResNeXt101 combined with augmented data. A key gap was the limited diversity of GAN outputs and their generalization challenges. In conclusion, augmentation was the most reliable and scalable solution for class imbalance, while GAN-based synthesis has potential but needs more improvement.

Y. Kuang [12] introduced PiCCL, a lightweight multiview contrastive learning

framework designed for efficient self-supervised training. Unlike traditional methods, PiCCL used a multiplex Siamese architecture with multiple identical branches (P=4 or P=8) and also introduced PiCLoss, a new loss function that averages embeddings to avoid quadratic scaling. The framework created multiple augmented views of each input to enhance representation learning. On benchmark datasets (CIFAR-10, CIFAR-100, and STL-10), PiCCL achieved the best accuracies of 94%, 72%, and 97%, respectively, and maintained 93% accuracy on STL-10 even with a batch size of eight. This performance surpassed other self-supervised methods by about 3%. The best results were achieved with multi-view embedding using PiCLoss. However, PiCCL has not yet been tested on domain-specific or agricultural datasets. In conclusion, the framework showed that self-supervised learning can be achieved with low computational cost, which makes it suitable for edge applications in agriculture.

Y. Wang et al. [13] advanced self-supervised learning for plant disease recognition by combining masked autoencoders (MAE) with convolutional block attention modules (CBAM). Two datasets were used: CCMT with 88,010 field images of cashew, cassava, maize, and tomato divided into 18 classes, and a custom potato dataset with 3,256 images of healthy, early blight, and late blight categories. Data splits followed standard train, test, and validation ratios with extensive augmentation applied. The MAE+CBAM model was compared with ResNet-50, ViT, and vanilla MAE. Results showed that MAE+CBAM achieved the highest accuracy of 95.35% on CCMT and 99.61% on the potato dataset, also achieving F1-scores of 95.52% and 98.62%, respectively. There were consistent improvements of 1–2% over the baselines. However, it was not well tested for different crops or long-term new conditions. In conclusion, the study showed that SSL can reduce the need for labeled data and can be used in real field systems.

Kiran. S. M. et al. [14] developed a classical machine learning pipeline for plant leaf disease detection, focusing on efficient preprocessing and handcrafted features. The dataset combined PlantVillage (38 classes) with MangoLeafDB (8 classes), resulting in 41,546 images across 40 categories, which were split into an 80:20 ratio for training and testing. Preprocessing methods included resizing, histogram equalization, Gaussian denoising, and segmentation to isolate leaves from background clutter. Features such as Hu moments, Haralick textures, and color histograms were extracted, and models including logistic regression, LDA, KNN, Decision Tree, Random Forest, and SVM were evaluated. Among these models, Random Forest achieved the highest accuracy of 97.92%. Segmentation improved accuracy by 2.19%, and hyperparameter optimization contributed an additional 0.48% accuracy. Limitations are included vulnerability to noisy field conditions such as occlusion and lighting variations. But this approach struggles in real-world conditions where leaves may be blocked or lighting changes. Still, the results prove that traditional machine learning paired with good preprocessing can perform well

without needing heavy computing resources.

H. Ghosh et al. [15] conducted a comparative benchmarking study on mango leaf disease detection, applying machine learning models with deep learning architecture. The dataset contains 5000 mango leaf images evenly distributed to 8 classes, where seven disease classes and one healthy class were categorized, with 500 images per class, which were split into 70% training,15% validation, and 15% test sets. Data was augmented using rotation, flipping, zoom, and brightness adjustments. Machine learning models such as K-Nearest Neighbors and Random Forest, as well as deep learning models including AlexNet, DenseNet21, ResNet50, and EfficientNet-B0 were applied. After testing, KNN performed poorly with 53.1%, Random Forest achieved 84.4%, while AlexNet, DenseNet121, and ResNet50 achieved 95.1%,86.4%, and 92.1% respectively. But EfficientNet-B0 outperformed all others, achieving a 99.9% accuracy with a near perfect F1 score in all classes, which clearly showcases the advantages of deep learning over traditional machine learning models. Limitations include the lack of cross-orchard testing and limited interpolability of CNN decisions. In summary, this study established Efficient-B0 as a great model for mango leaf detection while highlighting the need for explainable models and wider validation.

Table 2.1: Summary of Literature Reviewed.

Author (s)	Year	Title	Methodologies	Key Findings
A. Katumbo et al. [1]	2024	Utilizing edge computing and deep learning for the real-time identification of bean plant diseased.	Experimental and Quantitative Analysis	Accuracy ~90%; YOLOv8 mAP@50 = 87.6%; deployed on edge devices.
M. V. Shewale et al. [2]	2023	High performance deep learning architecture for early detection and classification of plant leaf disease	Experimental and Quantitative Analysis	>99% accuracy; effective for early detection & preventing crop loss.
K. M. Hosny et al. [3]	2023	Multi-Class Classification of Plant Leaf Diseases Using Feature Fusion of Deep CNN and Local Binary Pattern	Experimental and Quantitative Analysis	Accuracy 99–99.6%; handles visually similar diseases better than CNN alone.

E. Moupojou et al. [4]	2023	FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification with Deep Learning	Dataset Development and Quantitative Benchmarking	Accuracy dropped to 65–77% in field conditions. Lightweight models with strong augmentation performed best.
M. O. Ojo and A. Zahid [5]	2023	Improving Deep Learning Classifiers Performance via Preprocessing and Class Imbalance Approaches in a Plant Disease Detection Pipeline	Experimental and Quantitative Analysis	Best: ResNet-50 + CLAHE + sharpening + GAN, 97.69% accuracy. Preprocessing & balancing improved results.
M. Alsaidi et al. [6]	2024	Tackling the class imbalanced dermoscopic image classification using data augmentation and GAN	Experimental and Quantitative Analysis	Augmentation reliable; ResNeXt101 + augmented data 93.2% accuracy. GAN is less effective.
Y. Kuang et al. [7]	2025	PiCCL: A lightweight multiview contrastive learning framework for image classification	Experimental and Quantitative Analysis	Accuracy: 94% (CIFAR-10), 72% (CIFAR-100), 97% (STL-10). Outperformed other SSL by ~3%.
Y. Wang et al. [8]	2024	Classification of Plant Leaf Disease Recognition Based on Self-Supervised Learning	Experimental and Quantitative Analysis	High accuracy: 95.35% (CCMT), 99.61% (potato). F1: 95.52%, 98.62%. SSL reduces the need for labeled data.
Kiran. S. M. et al. [9]	2023	Plant Leaf Disease Detection Using Efficient Image Processing and Machine Learning Algorithms	Experimental and Quantitative Analysis	Random Forest: 97.92% accuracy. Preprocessing + tuning improved results.
H. Ghosh et al. [10]	2024	Benchmarking ML and DL Models for Mango Leaf Disease Detection: A Comparative Analysis	Comparative Analysis	EfficientNet-B0 best: 99.9% accuracy, near-perfect F1. DL outperformed traditional ML

2.2.1 Similar Research

D. Torpey and R. Klein [16] addressed the difficulty of applying self-supervised learning (SSL) to 3D medical images, where most approaches depended on converting 2D models into 3D versions. While this strategy was effective, 3D models were extremely costly in terms of computation, which made them less practical for many researchers. The authors introduced DeepSet SimCLR, a framework that built on 2D SimCLR models but was still able to handle 3D medical data without the heavy costs of full 3D methods to overcome this issue. The framework also included two simple lightweight models to keep the models compact and efficient while adding little parameter complexity. Although data-loading costs increased, these could be resolved through parallaly. The authors explained that their method avoided the heavy architectural modifications usually required in existing 3D models that makes it easier to train while still keeping strong representation quality. The result showed that both proposed methodologies consistently outperformed the baseline SimCLR models in different condition tasks. These included tasks such as classification and transfer learning, where the new models showed stronger generalization compared to the baseline. They showed well performance when camparing with more advanced 2D and 3D existing models. The study also showed that the achievement in efficiency did not come with the loss of accuracy, as the models continued to perform strongly through different evaluations. This work demonstrated that SSL could be both easy and effective to use and making it more useful for researchers with limited computing resources. DeepSet SimCLR could deliver both performance and availability at the same time with this contribution and creating new possibilities for using SSL in 3D medical imaging.

M. Das and Z. Zhong [17] studied the limits of current self-supervised learning (SSL) methods, which mostly used common data augmentations such as cropping, flipping, and color changes. While these helped in training, they did not provide enough variety, which reduced the robustness of SSL models. The authors proposed ViewMix which is a simple augmentation method to address this problem that produced mixed views by combining patches from different images. This approach gave the model a wide range of training examples and helped it to learn more general and stable features. ViewMix was easy to apply, required no extra parameters, and worked smoothly with existing SSL methods like SimCLR and BYOL without any extra cost. ViewMix consistently improved results on their experiment. SimCLR gained 1.20%, VICReg improved by 1.74%, and VIbCReg by 2.34% in top-1 accuracy and combining it with VICReg improved performance by 3.24%. Translation and Rotation increased accuracy by 6.63% under robustness testing with unseen transformations. Models also achieved higher F1-scores and showed greater stability during training. This work improved the strength of SSL by increasing accuracy, improving F1-scores, and increasing robustness with a lightweight design. This contribution provided a simple but effective way to improve SSL frameworks and ensure more stable model performance.

Huan et al. [18] explored a unique self-supervised learning (SSL) framework for plant disease detection, reducing the need for large labeled datasets, which are costly in the agricultural field. The framework integrates BYOL, MIM, and contrastive learning on a ResNet101 backbone, by using a hybrid loss that balances semantic alignment, image redesigning, and occurrence separation. GPU-based augmentation improved robustness by introducing random variations during pretraining. After evaluation on PlantDoc, the model obtained 77.82% accuracy and a 77.48% macro F1 score which performing as well as or better than supervised approaches. When fine-tuned on PlantVillage, it achieved 99.85% accuracy which showed strong adaptability on datasets. Grad-CAM and t-SNE further confirmed that the model effectively identified disease relevant areas without any label. The combined SSL design, GPU-based augmentation, and validation of label efficiency are the main contributions. This work provides valuable scenario into building efficient and label-saving models for precision agriculture. SSL is shown as a scalable and practical tool for plant disease detection in both laboratory and field settings.

Ogidi et al. [19] a benchmarking study is presented of self-supervised contrastive learning (SSL) methods for plant image analysis that focusing on tasks such as wheat head detection, plant disease detection, spikelet counting, and leaf counting. The researchers tested two SSL methods, Momentum Contrast v2 (MoCo v2) and Dense Contrastive Learning (DenseCL). They compared them with traditional supervised pretraining. They used different datasets with including ImageNet, iNaturalist, iNaturalist Plants, and TerraByte Field Crop (TFC). The results showed that supervised pretraining usually worked better than SSL, except for leaf counting. But SSL performed almost as well. DenseCL was effective for the tasks that require detailed spatial information and showing that preserving spatial features is important in plant analysis as well. The study also found that SSL methods are more affected for repeated images in the training dataset, which is common in farm images where plants look very similar. The work showed that MoCo v2 and DenseCL learn similar features, but these are quite different from those learned by supervised models in deeper layers. The main contributions of this research are that it is the first complete benchmark of SSL for plant phenotyping, it studies how different datasets affect performance, and it examines the effect of repeated images on feature learning. The study showed both the advantages and challenges of using SSL for plant analysis and offers useful guidance for future research.

Dong et al. [20] studied self-supervised learning (SSL) for maize kernel classification and embryo segmentation. These tasks help to predicted oil content and improve seed sorting in breeding programs. The work used SSL methods called SimCLR and NNCLR with a ResNet50 model. The models were first trained on unlabeled images and then fine-tuned for classification and segmentation. The results showed SSL-pretrained models worked better than ImageNet-pretrained or randomly initialized models. The researchers got a Dice score of 0.81, compared to 0.78 and 0.73. Strong results were possible even with only 1% of labeled data, showing SSL can save labeling work. One SSL-pretrained model also worked well for both tasks, showing it can be reused. The main contributions are

creating a full pipeline for kernel analysis, showing that SSL which can save labeling, and reducing the need for expert labels. The study shows SSL is useful for plant science and can help with breeding and production tasks

2.3 Gap Analysis

The survey of previous research shows that CNNs such as ResNet, DenseNet, and EfficientNet achieve strong results in plant disease classification, while hybrid approaches combining handcrafted features with CNN embeddings improve discrimination for visually similar diseases [16]. Augmentation strategies, GAN-based synthesis, and preprocessing methods like CLAHE have helped mitigate dataset limitations, and lightweight models demonstrate potential for real-time field deployment. However, major gaps remain: most studies depend heavily on augmentation or oversampling, which introduces synthetic artifacts and limits generalizability; supervised learning dominates, requiring large annotated datasets that are often unavailable in agriculture; and hybrid or edge AI approaches, though promising, remain dataset-specific and fail to fully address representation learning under imbalance. Emerging self-supervised learning (SSL) frameworks like PiCCL and MAE show promise but have not been adequately tested on agricultural datasets, especially for beans and bean leaves, and few works provide systematic benchmarking against classical methods. To fill these gaps, this research adopts a dual-pipeline strategy that combines SimCLR-based self-supervised learning with Sobel-Feldman and contour-based feature extraction, offering both modern and traditional baselines. This approach reduces reliance on augmentation, addresses imbalance without synthetic oversampling, and establishes a comprehensive benchmark for advancing plant disease classification in low-resource environments.

Table 2.2: Gap Analysis Table

Features	Dataset Used	Approach / Model	Strengths	Limitations	Identified Gap	Relevance to Current Study
CNN features	Bean plant field dataset	ResNet, MobileNet on edge devices	Low latency, edge deployment	Focused only on interface speed; dataset limited	Did not address imbalance or SSL	Highlights real-time deployment but lacks SSL
CNN features + Augmentation	PlantVillage (multiple crops)	Fine-tuned CNN with residual connections	High accuracy (99%+)	Relied heavily on augmentation; dataset not realistic	Dependency on synthetic augmentation	Shows CNN success but generalization issue
CNN + LBP (texture)	3,171–3,948 leaf images	Hybrid CNN + LBP features	Improved for similar diseases	Single dataset; no field validation	Dataset specificity, no SSL	Motivates hybrid but shows lack of robustness
CNN features	FieldPlant dataset (5,170	MobileNet, EfficientNet, YOLO	Realistic dataset, IoT support	Performance dropped in field (65–	Realism challenge unsolved	Supports testing on real-world

	images)			77%)		data
CNN + GAN/Augmented features	Tomato leaf greenhouse images	ResNet, Xception, GAN augmentation	Improved accuracy with preprocessing + GAN	Tested only in greenhouse	Over-reliance on augmentation + GAN	Gap: imbalance not solved in real field
CNN + GAN-generated features	HAM10000 (medical dataset)	VGG, ResNet, GAN augmentation	Showed impact of GANs on imbalance	GAN samples lacked diversity	GAN less stable than augmentation	Relevance: imbalance problem parallels beans
SSL contrastive features	CIFAR-10/100, STL-10	PiCCL (SSL framework)	Lightweight, efficient SSL	Not applied to agriculture	SSL unexplored in crops	Direct link to our SSL focus
SSL embeddings + attention	CCMT (88k), Potato dataset (3,256)	MAE + CBAM (SSL)	Improved accuracy, less labeling	Not tested widely across crops	SSL needs domain-specific testing	Supports SSL use in agriculture
Haralick, Hu Moments, Color	PlantVillage + MangoLeafDB	Traditional ML + handcrafted features	High accuracy with RF (97.9%)	Vulnerable to noise, occlusion	Classical ML limited in field use	Relevant for benchmarking
CNN + Augmented features	Mango leaf dataset (5000 images)	CNNs vs ML (EfficientNet best)	EfficientNet 99.9% accuracy	Lacked cross-orchard validation	Limited generalization	Shows DL dominance but lacks robustness
CNN features + augmentation	PlantVillage, curated datasets	VGG, ResNet, DenseNet	High lab accuracy	Drop in noisy field data	Supervised reliance, poor generalization	Motivates SSL use
CNN + LBP, Haralick, etc.	Single crop datasets	CNN + handcrafted features	Improved discrimination	Dataset-specific	Lack of transferability	Benchmarks classical + hybrid methods
Augmented synthetic features	Various crops	SMOTE, GAN, heavy augmentation	Improves balance	Synthetic artifacts, limited realism	Augmentation dependency	Gap targeted by SSL
SSL embeddings + edge features	Bean/Leaf dataset (8 classes)	SimCLR + Sobel-Feldman + CNN/ML	Addresses imbalance w/o SMOTE; dual pipeline	Limited crop scope	Few SSL + classical benchmarks exist	Fills benchmarking and imbalance gap

Table 2.2 highlights the key limitations in current data imbalance problems and proposes targeted methodologies to bridge the gaps, thereby contributing to the development of more robust and agriculturaly effective.

2.4 Summary

In summary, existing studies on plant disease classification have achieved high accuracy using CNNs, hybrid models, and augmentation techniques, yet they remain limited by heavy reliance on synthetic data, large annotated datasets, and poor generalization in real-world field conditions. While self-supervised learning frameworks have recently emerged as a solution, their application to agricultural datasets—particularly beans and bean leaves—remains underexplored. Moreover, few works provide systematic benchmarking between modern SSL methods and traditional baselines. This research addresses these gaps through a dual-pipeline approach that combines SimCLR with Sobel-Feldman feature extraction, reducing dependence on augmentation and improving robustness under class imbalance.

Chapter 3

Research Methodology

Research Methodology represents a well-structured outline adopted for developing an automated bean and bean leaf disease classification system using a dual-pipeline approach. It explains the general design of the project, which combines self-supervised learning through SimCLR with classical image processing techniques for benchmarking comparison. This chapter further details the requirement specifications, system design, data flow, and evaluation strategy, along with the overall project plan. These elements provide a structural foundation for implementing the work in line with engineering standards and scientific practices.

3.1 Methodology

The methodology of this research is carefully designed to evaluate the effectiveness of self-supervised learning for agricultural image classification also by providing comparative insights against traditional feature-based machine learning approaches. A dual-pipeline framework adopted in the study by ensuring a complete analysis that addresses the challenges of class imbalance, dependency on augmentation, and the need for lightweight but accurate models. This section outlines the full methodology, system design specifications, functional and nonfunctional requirements, data flow, and interface considerations that form the foundation of the research.

3.1.1 Overview

The target of this research is according to the design of evaluating the effectiveness of self-supervised learning in addressing class imbalance and data scarcity challenges for agricultural image classification. A dual-pipeline strategy is adopted the first pipeline leverages self-supervised representation learning with SimCLR using deep learning backbones, while the second employs classical image processing techniques such as Sobel-Feldman edge detection and contour analysis with machine learning classifiers. Both pipelines are applied to the same dataset of bean and bean leaf images, ensuring that results are comparable and reflective of real-world performance under different methodological models.

This chapter provides,

i. An overview of the adopted methodologies

ii. The design specifications

iii. Functional and nonfunctional requirements of the system

iv. Data flow and user interface design considerations.

Together, these elements establish a structured foundation for the research and guide the implementation described in subsequent chapters.

3.1.2 Proposed Methodology

The proposed methodology employs a dual-pipeline framework designed to investigate the effectiveness of self-supervised learning in comparison with traditional feature-based approaches for the classification of bean and bean leaf diseases. The first pipeline is focused on self-supervised learning using SimCLR, where the backbone model is based on ResNet18, and DenseNet121 deep learning model. The second pipeline focused on classical feature extraction techniques, including Sobel-Feldman edge detection and contour analysis which is followed by lightweight machine learning classifiers like XGBoost. Both pipelines are trained and evaluated on the same dataset containing eight classes, what ensuring a consistent basis for comparison.

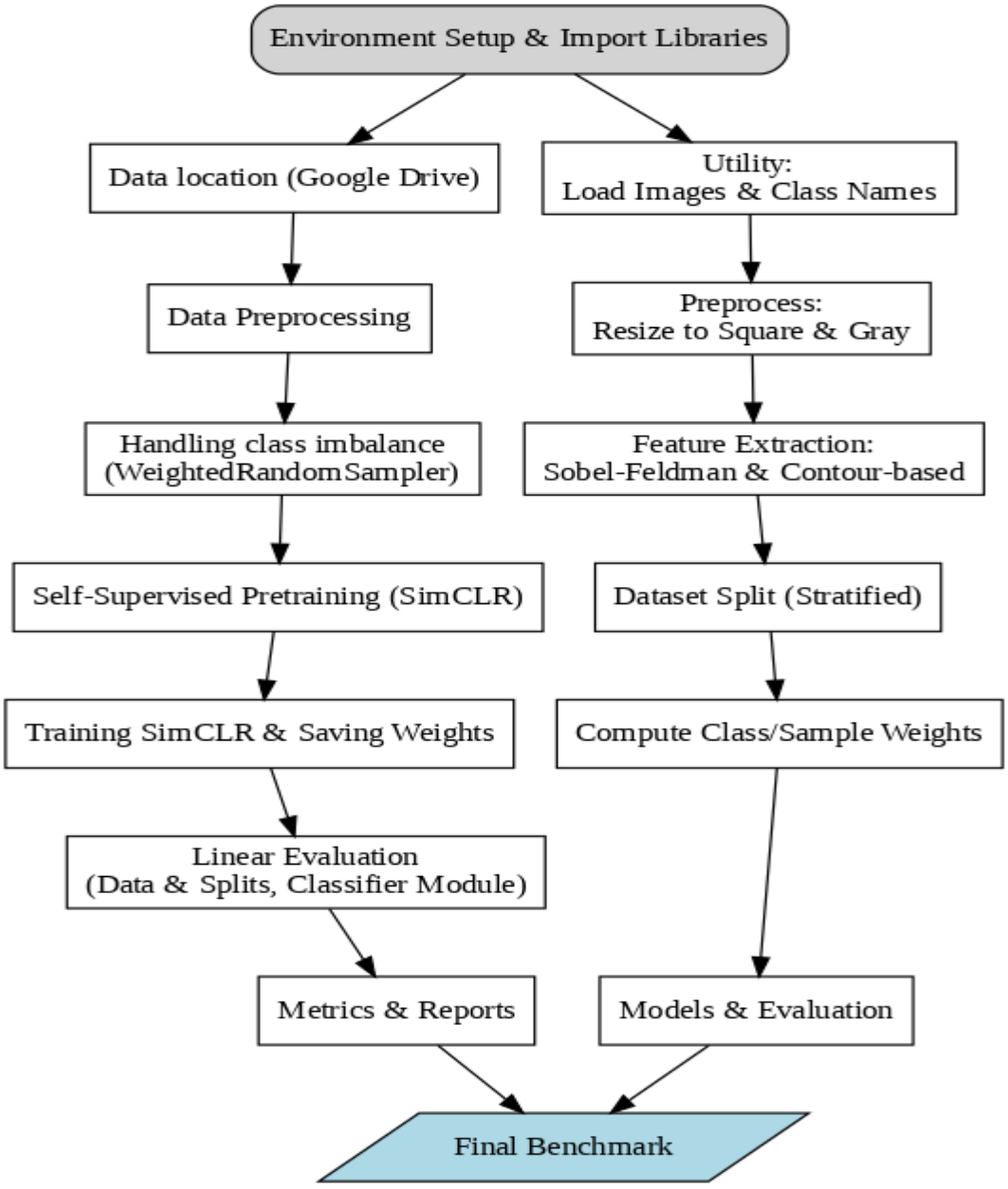


Figure 3.1: Workflow Diagram

i. **Data Preprocessing:** The dataset contains the images that classified into eight different categories where six classes of bean leaves and two classes of beans. Before training, the images are standardized through resizing, normalization, and format conversion to ensure compatibility with both deep learning and traditional feature-extraction pipelines. A set of light augmentations for SimCLR pretraining such as random cropping, horizontal flipping, grayscale conversion, Gaussian blur, and color jittering are applied to create multiple views of each image. These transformations are intentionally kept minimal, as the main goal of SimCLR is to learn meaningful representations without relying heavily on handcrafted augmentation schemes.

ii. **Handling Class Imbalance:** The dataset is inherently imbalanced, with some classes containing as few as 60 images compared to over 1,000 in others. Instead of using oversampling methods like SMOTE, this research leverages sample weighting and WeightedRandomSampler to ensure that underrepresented classes contribute equally during training. In the self-supervised pipeline, SimCLR naturally mitigates imbalance by focusing on representation learning from all samples irrespective of their frequency. In the ML pipeline, class weights and inverse-frequency weighting are applied during training of models such as Random Forest and XGBoost to reduce bias toward majority classes.

iii. **SimCLR Method:** SimCLR (Simple Framework for Contrastive Learning of Visual Representations) forms the backbone of the self-supervised pipeline [17]. A ResNet18 encoder is employed to project images into a latent feature space, followed by a projection head that maps them into a lower-dimensional contrastive space. Through the NT-Xent loss, the framework maximizes agreement between augmented views of the same image while ensuring separation from representations of different images. This contrastive learning approach yields highly discriminative feature embeddings [18]. After pretraining, the backbone is frozen and fine-tuned with a linear classifier for downstream classification, allowing the learned representations to be evaluated in a supervised setting. For comparison and scalability, DenseNet121 is also explored as an alternative deep learning backbone.

iv. **Linear Evaluation:** To assess the quality of the self-supervised features, a linear classifier is trained on top of the frozen backbone. This step provides insight into how well SimCLR captures meaningful structures in the data and how effectively it reduces reliance on large labeled datasets. Evaluation metrics such as accuracy and F1-score are computed during this stage to benchmark performance.

v. **Sobel-Feldman and Contour Detection:** The classical pipeline is based on handcrafted feature extraction. Using the Sobel-Feldman operator, gradient-based features are extracted from grayscale images, capturing directional changes and edge structures [19]. Contour detection is applied using Canny edge detection, providing additional shape-related descriptors such as contour counts, areas, and perimeters [20]. These structural features are complemented by HSV color histograms and GLCM-based

texture features, ensuring that both visual appearance and spatial structure are represented.

vi. **Model Evaluation (DL and ML):** For the deep learning pipeline, ResNet18 and DenseNet121 models are pretrained with SimCLR and then fine-tuned with a linear evaluation head. Their performance is validated using classification reports, confusion matrices, and ROC curves across the eight classes. In the machine learning pipeline, extracted features are used to train Decision Tree, Random Forest, and XGBoost classifiers. Evaluation includes per-class precision, recall, F1-scores, as well as overall accuracy, supported by confusion matrix visualizations.

Figure 3.1 describes that the combination of these two pipelines, the methodology highlights the comparative strengths of modern self-supervised learning against traditional feature-engineering-based approaches. SimCLR demonstrates the ability to handle limited data and imbalance through robust representation learning, while the classical pipeline provides interpretable baselines, ensuring the study contributes a holistic evaluation of disease classification methods for beans and bean leaves.

3.1.3 Functional and Nonfunctional Requirements

A clear understanding of the system's functional and nonfunctional requirements is crucial for guiding the design and development of the Bean and Bean Leaves disease classification.

Functional Requirements:

i. Dataset Preparation: The system must allow loading and preprocessing of images across eight classes divided into beans and bean leaves.

ii. Dual-Pipeline Implementation: The system must implement both the self-supervised SimCLR-based pipeline and the classical feature-engineering-based pipeline for comparative benchmarking.

iii. Class Imbalance Handling: Weighted sampling or inverse-frequency weighting must be integrated to reduce the impact of class imbalance during training.

iv. Feature Extraction: The system must be able to extract edge-based, color, and texture features (Sobel, contours, HSV histograms, and GLCM) for use in traditional ML models.

v. Model Training and Evaluation: The framework must support training deep learning models (ResNet18, DenseNet121) and machine learning classifiers (Decision Tree, Random Forest, XGBoost), with detailed evaluation metrics including accuracy, F1-score, precision-recall, confusion matrices, and ROC curves.

vi. Cross-Validation Support: The system must allow k-fold cross-validation to ensure generalizability of the results, particularly for the ML classifiers.

vii. Result Visualization: Graphical outputs such as confusion matrices, ROC curves, and comparative performance plots must be generated for analysis.

Nonfunctional Requirements:

i. Scalability: The methodology should be adaptable to larger datasets and different plant species beyond beans.

ii. Efficiency: Training and feature extraction must be optimized to run within available computational resources such as Google Colab's GPU and standard desktop environments.

iii. Robustness: The system should provide consistent performance despite dataset imbalance or variation in image quality.

iv. Interpretability: Outputs from classical models must remain interpretable to support decision-making in agricultural contexts, while deep learning results should be complemented with visualization techniques such as classification report confusion matrix.

v. Reproducibility: All processes, including preprocessing, training, and evaluation, should be reproducible using fixed random seeds and documented experimental settings.

3.1.4 Data Flow Diagram

The data flow of the research follows the structured sequence like dataset input → preprocessing → pipeline split → model training → evaluation → final results that shown in the Figure 3.2

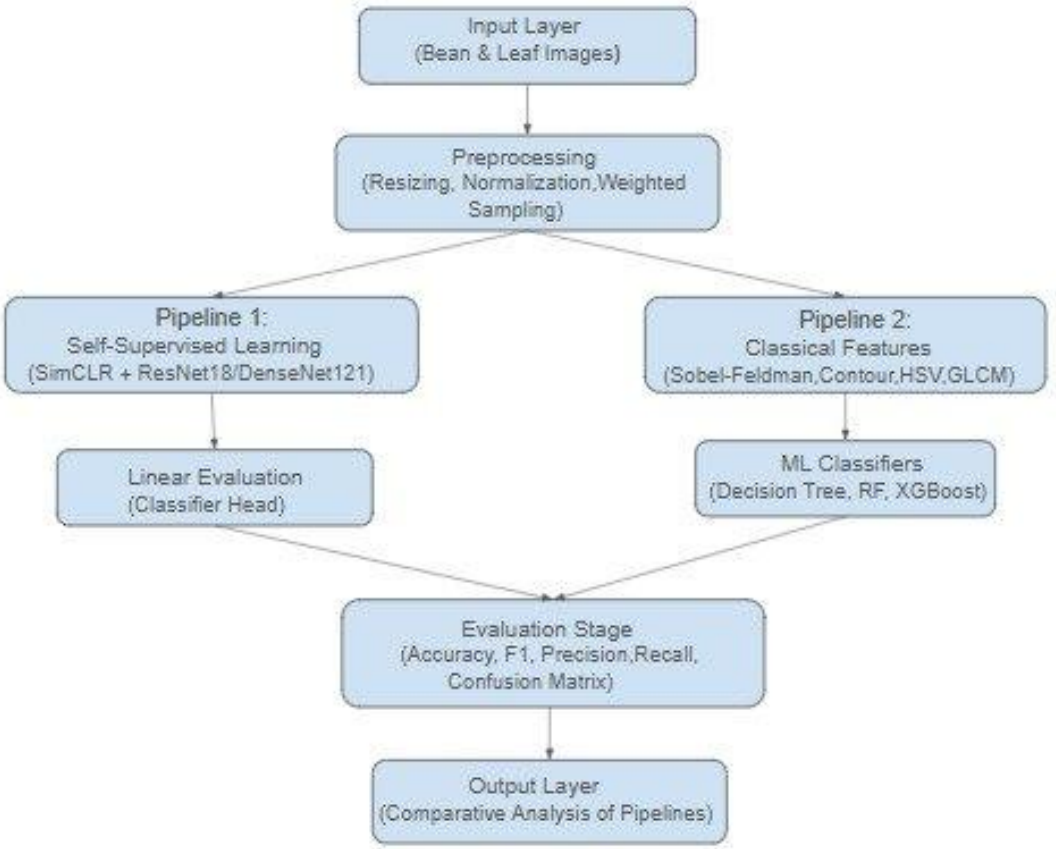


Figure 3.2 Data Flow Diagram

i. Input Layer – Raw bean and bean leaf images are collected as input.

ii. Preprocessing Stage – Images are resized, normalized, and transformed for consistency. At this stage, weighted sampling or inverse-frequency weighting is applied to address imbalance.

iii. Pipeline Split – The data flows into two distinct pipelines:

- Pipeline 1 (Self-Supervised DL): Images undergo light augmentations and are fed into SimCLR with ResNet18 or DenseNet121 backbones. Feature representations are learned in a contrastive manner, followed by a linear classifier for final predictions.
- Pipeline 2 (Classical ML): Images are processed to extract Sobel-Feldman, contour, HSV histogram, and GLCM texture features. These handcrafted features are then passed into ML classifiers (XGBoost).

iv. Evaluation Stage – Both pipelines produce classification results that are compared using accuracy, F1-score, and precision-recall analysis. Confusion matrices and ROC curves are generated to provide visual insights.

v. Output Layer – Comparative performance analysis is produced, highlighting the contributions and limitations of each approach.

3.1.5 UI Design

Although the original scope of this project was primarily focused on developing and testing backend models for bean and bean leaf disease classification, significant emphasis was also placed on designing and implementing a user-friendly graphical user interface (GUI). The result is a complete and interactive web-based system that enables farmers, agricultural specialists, and researchers—particularly those without a technical background—to easily access and utilize automated plant disease detection.

The current interface provides an intuitive and responsive platform where users can upload bean or bean leaf images and instantly receive classification results. Key features of the UI include a streamlined image upload field, real-time output visualization of the predicted disease class along with confidence scores, and the ability to reset and test new images seamlessly. The frontend has been developed using modern web technologies such as HTML5, CSS3, and JavaScript, and is integrated with the DRF backend through RESTful APIs to perform inference.

To ensure accessibility and usability across a wide range of devices, the interface follows responsive design principles and prioritizes user experience. While the core functionalities are already implemented, the system has been structured to support future enhancements such as batch image uploads, user authentication, result analytics, and downloadable diagnostic reports, making it scalable for farm-level deployment, research applications, and broader agricultural use.

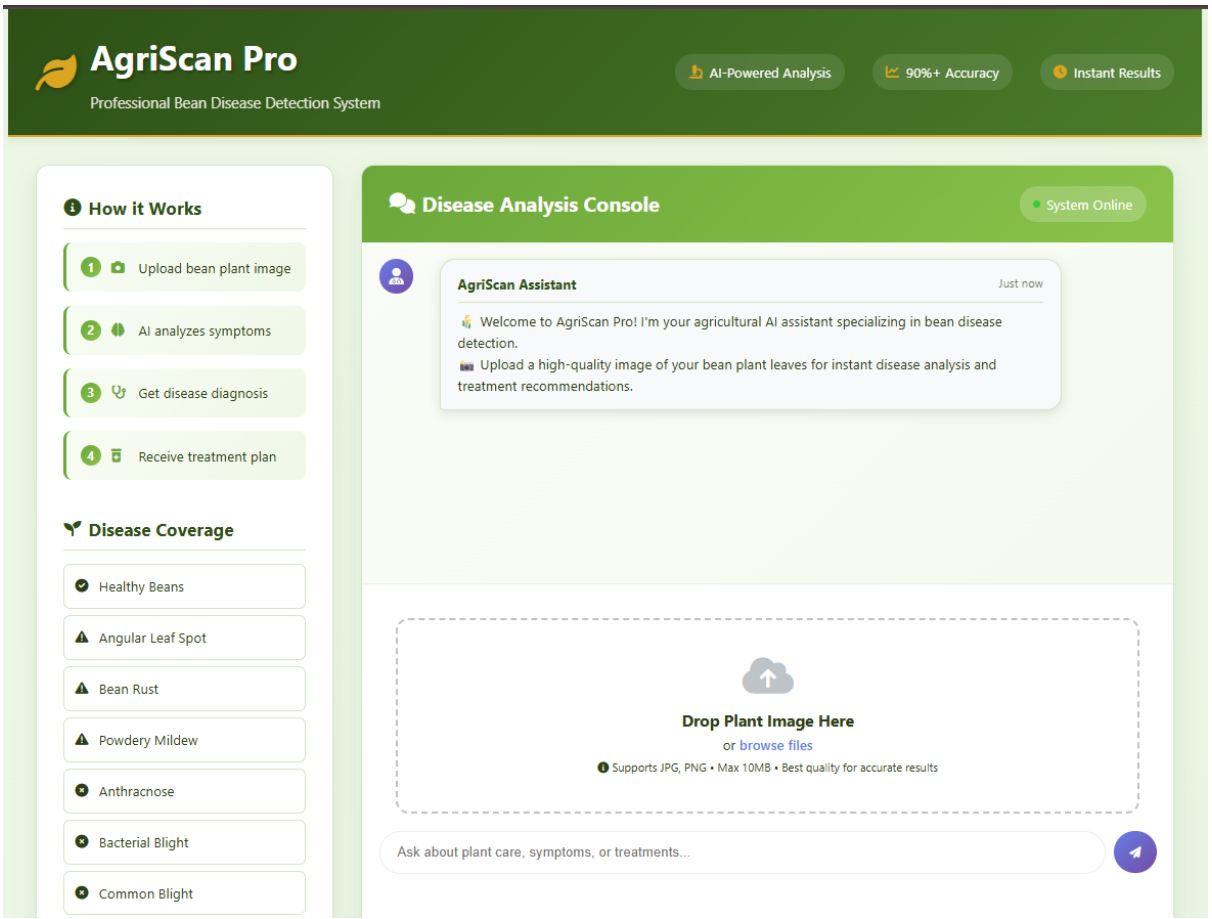


Figure 3.3 UI Interface of Disease Classification

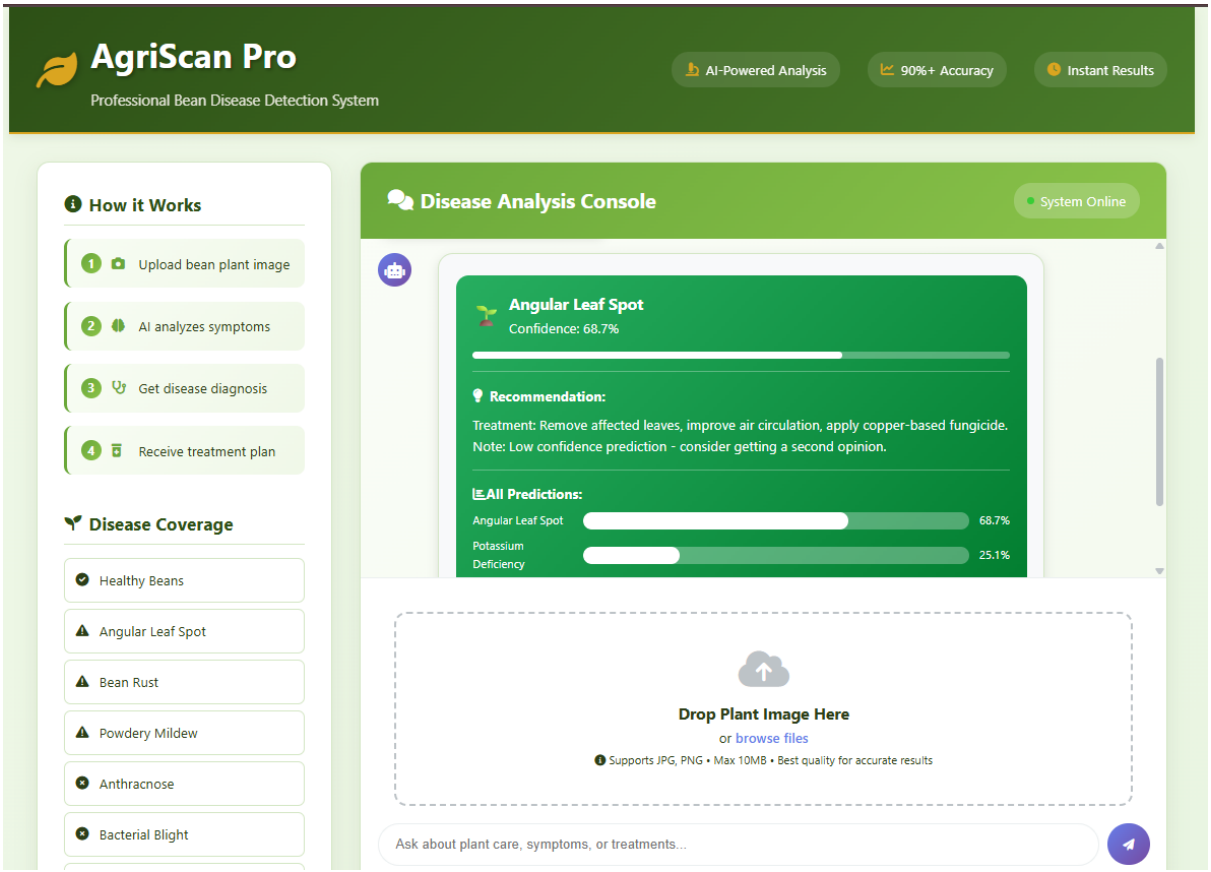


Figure 3.4 Output Section of the Disease Classification System

Figure 3.3 and 3.4 displays this UI serves as an essential bridge between advanced AI capabilities and real-world agricultural applications, enabling broader adoption and impact of the bean and bean leaves disease classification.

3.2 Detailed Methodology and Design

The detailed methodology expands upon the proposed system design by describing the sequential steps in depth. The dataset is first preprocessed to standardize size and format, with class imbalance addressed through sampling strategies. In the deep learning pipeline, SimCLR pretraining is conducted using ResNet18 and DenseNet121 backbones. After representation learning, the encoder is frozen, and a linear classifier is trained to perform supervised classification. For the classical pipeline, features derived from Sobel-Feldman edge detection, contour analysis, HSV histograms, and GLCM texture analysis are extracted and used to train XGBoost models. The evaluation stage involves both per-class and macro-level performance reporting, supplemented by visualization tools for clear interpretation.

1. Data Distribution

The dataset comprises eight classes, categorized into six bean leaf classes (Good Leaves, Potassium Deficiency, Halo Blight, Fungi Pathogens, Cercospora Leaf Spot, Angular Leaf Spot) and two bean classes (No Disease Bean, Bacterial Pathogen). The distribution is imbalanced, with Good Leaves having 1164 images, while Potassium Deficiency includes only 60. The visualization by a bar diagram has been shown in the Figure 3.3.

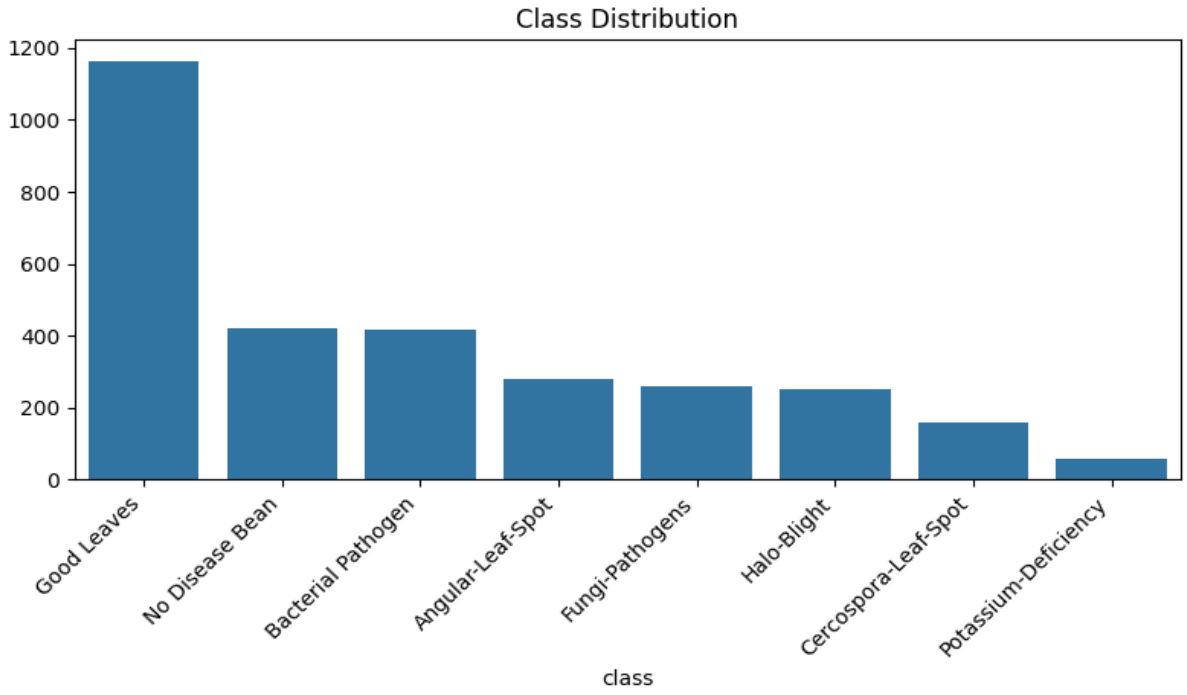


Figure 3.3 Data Distribution Bar Diagram

This skew necessitates careful handling during training, as models tend to bias towards majority classes. Instead of relying solely on synthetic oversampling such as SMOTE, the methodology leverages self-supervised representation learning to minimize imbalance effects.

2. Data Description and Preparation

The collected data were 3,011 images having 768 X 960pixel dimensions and 72 X 72 dpi image resolution. There are 8 different unique types of data along with bean and bean leaf with identification and labelling of different diseases. Collecting bean and bean leaf images is an old traditional way where plants are gathered from the bean cultivated land cultivators or farmers are growing to use their mitigation benefits. In the time of data collection, so many difficulties have been faced. Some people do not agree with us for capturing images into our devices. On the other hand, in the time of image capturing it should take risk to enter into the bean harvesting land because of many insects and other beasts. But still image collection has been successfully done. These images have descent qualities that have been used in detecting both diseased and diseased free beans and bean leaves. Some crucial disease types like Fungi Pathogens, Potassium Deficiency are included in our dataset which are absent in the others existing dataset. This dataset contains both good and bad images of the objects of 10 different classes as described below in Table the 3.1.

Table 3.1 Bean and Bean leaf image classes and number of images.

SL	Classes	No. of Images
1	Healthy-Leaf	1,164
2	Angular-Leaf-Spot	270

3	Cercospora-Leaf-Spot	160
4	Fungi-Pathogens Leaf	260
5	Halo-Blight	250
6	Potassium-Deficiency Leaf	60
7	Bacterial-Pathogen	420
8	No-Disease-bean	418

All images were first resized to a fixed resolution (224×224) to align with CNN backbone input requirements. Normalization was applied to scale pixel values between 0 and 1, ensuring stability in training. For the classical ML pipeline, preprocessing included conversion to grayscale, Sobel-Feldman edge detection, contour extraction, HSV histogram computation, and GLCM-based texture feature extraction. For the deep learning pipeline, augmentation was applied sparingly (rotations, flips) only during SimCLR pretraining, where contrastive learning benefits from view diversity. The prepared dataset was then split into training, validation, and test sets to facilitate unbiased evaluation.

3. Model Selection

For the self-supervised pipeline, the ResNet18 and DenseNet121 backbones were implemented as encoders within SimCLR. The encoders learn invariant feature representations via contrastive loss. After pretraining, the encoders are frozen and connected to a linear classifier, which is trained in a supervised manner on labeled data. This two-step process allows robust feature extraction without heavy dependence on labels.

For the classical pipeline features extracted through Sobel-Feldman and contour detection were fed into XGBoost models. XGBoost was chosen as the primary ML benchmark due to its scalability, regularization capabilities, and strong performance in tabular feature spaces.

4. Evaluation and Metrics

An extensive evaluation was carried out to measure the performance of the self-supervised and classical models. The evaluation was based on a range of automated metrics designed to capture different dimensions of classification performance, including overall accuracy, sensitivity to minority classes, and discriminative ability across multiple categories. Accuracy provided a general measure of correct predictions over the total samples, while precision and recall offered insight into how well the models balanced false positives and false negatives. The F1-score, particularly the macro-averaged F1, was emphasized to ensure that performance was not biased toward majority classes such as Good Leaves, but instead reflected fairness across underrepresented classes like Potassium Deficiency.

Confusion matrices were generated to further evaluate classification robustness, giving a detailed breakdown of misclassifications between similar disease categories. ROC (Receiver Operating Characteristic) curves and the Area Under the Curve (AUC) were used to assess how well the models discriminated between positive and negative samples for each class. These metrics provide a deeper understanding of model behavior beyond

simple accuracy, as they highlight both class-specific strengths and weaknesses.

Formally, precision and recall were computed as:

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

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

where TP denotes true positives, FP false positives, and FN false negatives. The F1-score was then derived as the harmonic mean:

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

For multi-class evaluation, the macro F1 was obtained by averaging the F1-scores of all classes equally that ensures minority categories were not overshadowed.

This comprehensive evaluation allowed for objective comparisons between self-supervised SimCLR-based representations and handcrafted feature-based ML classifiers. The results not only quantified model performance but also highlighted the strengths of SSL in handling imbalance without oversampling, while establishing baselines from classical approaches for benchmarking fairness.

5. Deployment

To demonstrate the model inference process for agricultural disease detection, we designed a lightweight deployment setup that can run efficiently on limited computational resources. The system was implemented with a simple user interface and a backend inference pipeline, ensuring that both deep learning and classical models could be tested in real-world scenarios.

i. Model Loading: The trained models, including SimCLR-based CNN backbones (ResNet18, DenseNet121) and ML classifiers such as XGBoost, are loaded into memory.

ii. Input Field: A file upload interface allows users to submit bean or leaf images captured from the field.

iii. Backend Inference: The backend processes the uploaded image through the selected pipeline—either the SSL-trained deep learning model or the handcrafted-feature-based ML model—and generates a classification output.

iv. Result Display: The predicted class (e.g., Good Leaf, Potassium Deficiency, Halo Blight, etc.) along with confidence scores is displayed on the interface, allowing users to immediately interpret the health condition of the bean or leaf.

This deployment setup ensures that the system remains accessible, lightweight, and

adaptable for use by farmers and researchers, even in resource-constrained environments.

3.3 Project Plan

The project was executed in a structured manner, divided into phases to ensure systematic development:

Phase 1 Literature Review and Problem Definition: Study of existing works on plant disease detection, self-supervised learning, and classical feature extraction to identify gaps.

Phase 2 Dataset Collection and Preparation: Organization of the bean and bean leaf dataset into eight classes, followed by preprocessing and imbalance handling.

Phase 3 Implementation of SimCLR Pipeline: Pretraining with SimCLR using ResNet18 and DenseNet121, followed by linear evaluation.

Phase 4 Implementation of Classical ML Pipeline: Feature extraction using Sobel-Feldman, contours, HSV, and GLCM, followed by training ML models (XGBoost).

Phase 5 Evaluation and Comparative Analysis: Performance evaluation of both pipelines using accuracy, F1-score, confusion matrices, ROC curves, and cross-validation for ML models.

Phase 6 Report Writing and Documentation: Compilation of findings into a structured report, including discussion, limitations, and future directions

The project was executed in a structured manner, divided into phases to ensure systematic development:

Phase 1 Literature Review and Problem Definition: Study of existing works on plant disease detection, self-supervised learning, and classical feature extraction to identify gaps.

Phase 2 Dataset Collection and Preparation: Organization of the bean and bean leaf dataset into eight classes, followed by preprocessing and imbalance handling.

Phase 3 Implementation of SimCLR Pipeline: Pretraining with SimCLR using ResNet18 and DenseNet121, followed by linear evaluation.

Phase 4 Implementation of Classical ML Pipeline: Feature extraction using Sobel-Feldman, contours, HSV, and GLCM, followed by training ML models (XGBoost).

Phase 5 Evaluation and Comparative Analysis: Performance evaluation of both pipelines using accuracy, F1-score, confusion matrices, ROC curves, and cross-validation for ML models.

Phase 6 Report Writing and Documentation: Compilation of findings into a structured report, including discussion, limitations, and future directions.

3.4 Task Allocation

This table shows a Gantt chart detailing the timeline of the principal activities in each period of the project from week 12 to week 48.

Tasks	Weeks																		
	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48
Data collection phase																			
Preprocess all the data																			
Model training																			
Create a demo application.																			

Four main tasks are listed along the vertical axis: Data collection phase, Preprocess all the data, Model training, and Create a demo application. The horizontal axis represents the project timeline in weekly increments. Each task is represented by a set of colored bars (blue and green), which indicate its duration.

The project appears to be sequential. The Data collection phase runs from week 12 to week 20. This is followed by the Preprocess all the data task, which begins around week 22 and ends around week 30. The Model training task then takes place from week 32 to week 40. Finally, the project concludes with Create a demo application, which spans from week 40 to week 48. While the chart clearly outlines the timeline and sequence of the tasks, the specific meaning of the blue and green bars is not provided, making it difficult to determine the exact nature of the two sub-phases within each task.

3.5 Summary

This chapter presented the methodological foundation of the research. A dual-pipeline approach was proposed, with one pipeline based on self-supervised learning using SimCLR and deep learning backbones, and the other based on traditional feature engineering and machine learning classifiers. Requirements were specified, the data flow and design principles were described, and the project plan was outlined. Together, these methodological steps provide a structured path for implementing the research and generating meaningful comparative insights between modern self-supervised techniques and classical baselines.

Chapter 4

Implementation and Results

This chapter presents the practical implementation of the proposed dual-pipeline framework and discusses the obtained results. The experiments were conducted using both the self-supervised learning (SimCLR) approach with deep learning backbones and the traditional machine learning pipeline based on handcrafted features. Each stage of the implementation, from environment setup to comparative evaluation, is described in detail.

4.1 Environment Setup

The experiments were carried out in a Python environment, applying both GPU-based and CPU-based resources. The primary implementation was managed on Google Colab, which provides entry to an NVIDIA Tesla T4 GPU for training the deep learning models. Development and debugging were performed using a Windows 11 machine (Intel Core i3 12th Gen, 8 GB RAM, 512 GB SSD) to ensure code portability across devices.

Key libraries and frameworks included PyTorch for implementing the SimCLR framework (simclr_backbone.py), scikit-learn for classical ML models (bean_ml.py), NumPy and Pandas for data handling, and Matplotlib/Seaborn for visualization. For preprocessing and augmentation, OpenCV and PIL were used, particularly for tasks such as resizing, normalization, grayscale conversion, and edge detection. The environment was configured by enabling seamless switching between deep learning and machine learning experiments, ensuring fair comparison across both pipelines.

4.2 Testing and Evaluation/Performance/ Comparative Analysis

The evaluation was conducted on the prepared dataset containing eight classes of beans and bean leaves. To ensure unbiased results, the dataset was split into training, validation, and test subsets, maintaining class distribution across splits.

For the self-supervised pipeline, SimCLR was first trained on the unlabeled dataset using ResNet18 and DenseNet121 as encoder backbones. During pretraining, the model learned discriminative representations by maximizing agreement between augmented views of the same image. These learned features were then fine-tuned with a linear classifier using the labeled portion of the dataset. Results demonstrated that SimCLR effectively reduced dependency on heavy data augmentation while capturing meaningful representations across all classes. The DenseNet121 backbone outperformed ResNet18 in overall accuracy and F1-score, particularly for minority classes like Potassium Deficiency.

Sobel-Feldman edge detection and contour-based features were extracted For the classical pipeline, followed by HSV histograms and GLCM-based texture features. These were evaluated using Decision Tree, Random Forest, and XGBoost classifiers. Among them, XGBoost consistently delivered the highest accuracy, owing to its ability to handle

nonlinear decision boundaries and avoid overfitting through regularization. While ML models performed adequately on majority classes, their performance declined sharply on minority categories due to limited feature separability.

The comparative analysis highlighted the strengths and weaknesses of both approaches. The SimCLR pipeline achieved superior overall accuracy and macro F1-scores, proving its ability to handle imbalance without relying on SMOTE or synthetic oversampling. The ML pipeline, on the other hand, offered interpretability and computational efficiency, making it suitable for environments with limited resources but less effective in terms of generalization.

Evaluation was supported by multiple performance metrics, including accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrices. The confusion matrices confirmed that SimCLR-based models were better at distinguishing visually similar classes, while ML classifiers often misclassified minority diseases into majority categories.

Table 4.1: Comparative Analysis

Dataset	Method	Model	Result
3011	SimCLR	DenseNet121	89.7%
3011	SimCLR	ResNet18	85.5%
3011	Sobel-Feldman	XGBoost	74.4%

The Table 4.1 compares the performance of three different machine learning and deep learning model and method combinations on a dataset of 3,011 items. The results are measured as a percentage, likely indicating classification accuracy. The most successful combination was the SimCLR method paired with the DenseNet121 model, achieving the highest result of 89.7%. This was followed by the SimCLR method used with the ResNet18 model, which produced a slightly lower result of 85.5%. The least effective combination was the Sobel-Feldman method used with the XGBoost model, which scored significantly lower at 74.4%. This comparison highlights the superior performance of the SimCLR self-supervised learning approach and the DenseNet121 deep learning architecture over the more traditional Sobel-Feldman and XGBoost combination for this particular classification task.

4.3 Results and Discussion

The results earned from both pipelines that presented which followed by a detailed discussion of their implications. The analysis focuses on comparing the performance of self-supervised learning models against traditional machine learning classifiers, with particular attention given to handling class imbalance, generalization across various categories, and robustness under limited data conditions. This section describes the strengths, limitations, and practical relevance of the proposed dual-pipeline framework by evaluating both quantitative metrics and qualitative patterns.

Table 4.2 Classification Report: (ResNet18)

	Precision	Recall	F1-score	Support
Angular Leaf Spot	0.84	0.91	0.88	280
Potassium Deficiency	0.80	0.50	0.62	60

Cercospora Leaf Spot	1.00	0.23	0.37	160
Fungi Pathogens	0.80	0.72	0.70	260
Good Leaves	0.85	0.89	0.87	1164
Halo Blight	0.81	0.84	0.71	250
No Disease	0.96	0.96	0.96	420
Bacterial Pathogen	0.95	0.94	0.94	417
Accuracy			0.85	3011
Macro Avg	0.88	0.75	0.76	3011
Weighted Avg	0.87	0.86	0.85	3011

Table 4.1 is a classification report, which summarizes the performance of a ResNet18 model across multiple classes. It presents four key metrics for each class: Precision, Recall, F1-score, and Support. Precision measures the accuracy of the positive predictions for each class. Recall measures the ability of model to find all positive instances of each class. F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. Support indicates the number of actual occurrences of each class in the dataset. The report also includes overall metrics. The Accuracy of the model is 0.85, or 85%, based on a total of 3011 samples. The macroAvg provides a simple average of the metrics over all classes, while the weightedAvg accounts for the number of samples in each class (support) when calculating the average. The model demonstrates strong performance for classes like "No Disease" and "Bacterial Pathogen," both of which have F1-scores of 0.94 and higher. In contrast, the model struggles with classes like "Cercospora Leaf Spot" and "Potassium Deficiency," which have low recall and f1-scores, indicating a significant number of misclassifications for these categories.

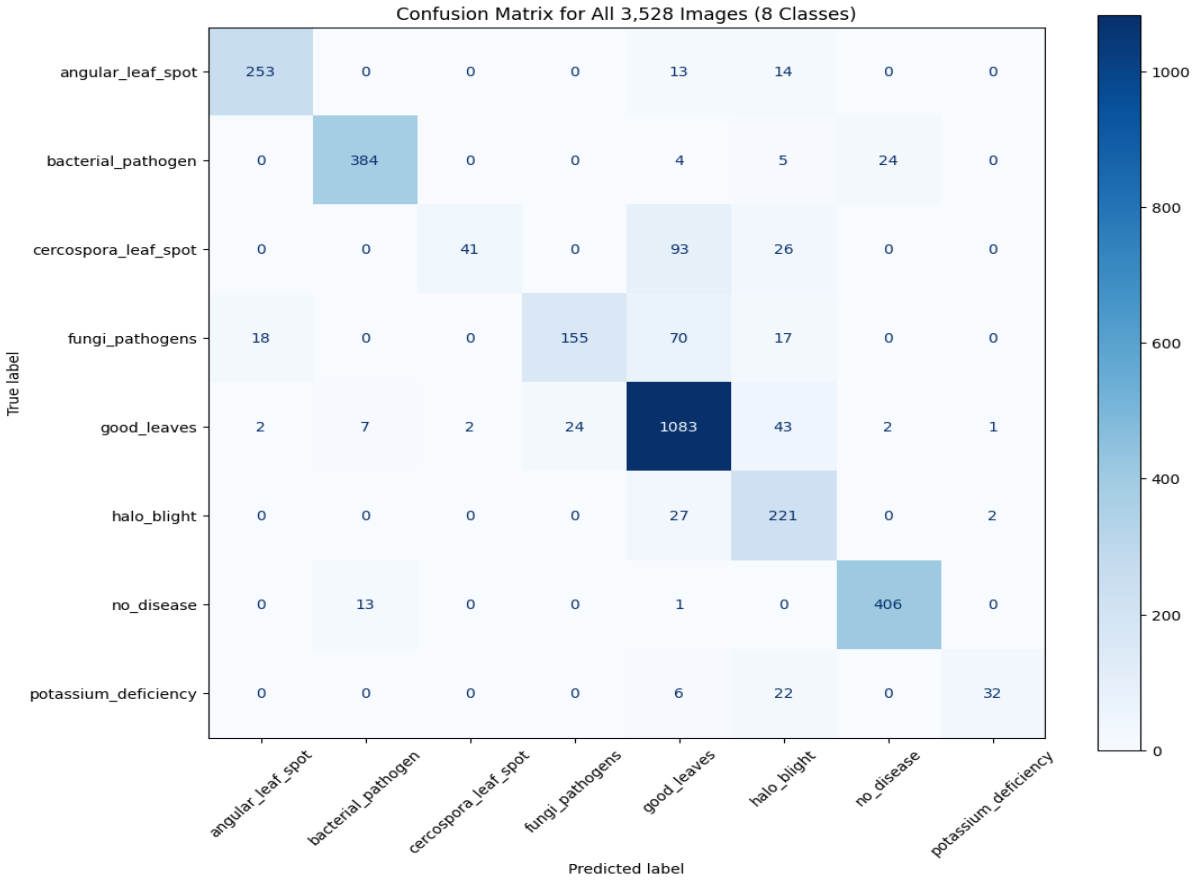


Figure 4.1 Confusion Matrix of ResNet Model

The Figure 4.3 displays a confusion matrix for a machine learning model, which has been used to classify 3,528 images into one of eight classes. The matrix evaluates the performance of the model by comparing the "True label" (the actual class of the image) with the "Predicted label" (the class the model assigned to the image). The classes are listed on both the vertical and horizontal axes and include various plant conditions such as "angular_leaf_spot", "bacterial_pathogen", "cercospora_leaf_spot", "fungi_pathogens", "good_leaves", "halo_blight", "no_disease", and "potassium_deficiency". The diagonal cells, displayed in darker shades of blue, show the number of correctly classified images for each category. For instance, the model correctly identified 1,083 images as "good_leaves". The off-diagonal cells represent misclassifications. For example, 18 images that were actually "fungi_pathogens" were incorrectly predicted as "angular_leaf_spot". Similarly, 13 images that were truly "no_disease" were misclassified as "bacterial_pathogen". The color bar on the right side provides a visual scale for the number of images, with darker blue representing a higher count. The matrix shows that the model performed well in identifying "good_leaves", "halo_blight", and "no_disease" but had some difficulty distinguishing between other classes.

Table 4.3 Classification Report: (DensNet121)

	Precision	Recall	F1-score	Support
Angular Leaf Spot	0.90	0.87	0.88	280
Potassium Deficiency	0.60	0.50	0.51	60
Cercospora Leaf Spot	0.76	0.70	0.73	160
Fungi Pathogens	0.88	0.85	0.86	260
Good Leaves	0.94	0.91	0.93	1164
Halo Blight	0.88	0.80	0.84	250
No Disease	0.97	0.98	0.98	420
Bacterial Pathogen	0.95	0.94	0.94	417
Accuracy			0.89	3011
Macro Avg	0.86	0.82	0.83	3011
Weighted Avg	0.92	0.89	0.90	3011

The provided table is a detailed classification report that summarizes the performance of a machine learning model. It presents key metrics, Precision, Recall, and F1-score, for eight distinct classes of plant conditions, as well as the Support (number of samples) for each class. The model achieves an accuracy of 89% on a dataset of 3011 samples. It performs exceptionally well on the "No Disease" and "Bacterial Pathogen" classes, with high F1-scores of 0.98 and 0.94, respectively. Performance is also strong for "Good Leaves" and "Fungi Pathogens," with F1-scores of 0.93 and 0.86. However, the model shows some difficulty with the "Potassium Deficiency" class, where it records the lowest F1-score of 0.51 and a recall of only 0.50, indicating that it struggles to identify all instances of this condition. The "Cercospora Leaf Spot" class also has a comparatively lower F1-score of 0.73. The "Weighted Avg" of 0.90 for the F1-score and the "Macro Avg" of 0.83 provide a comprehensive overview of the model's effectiveness across all classes.

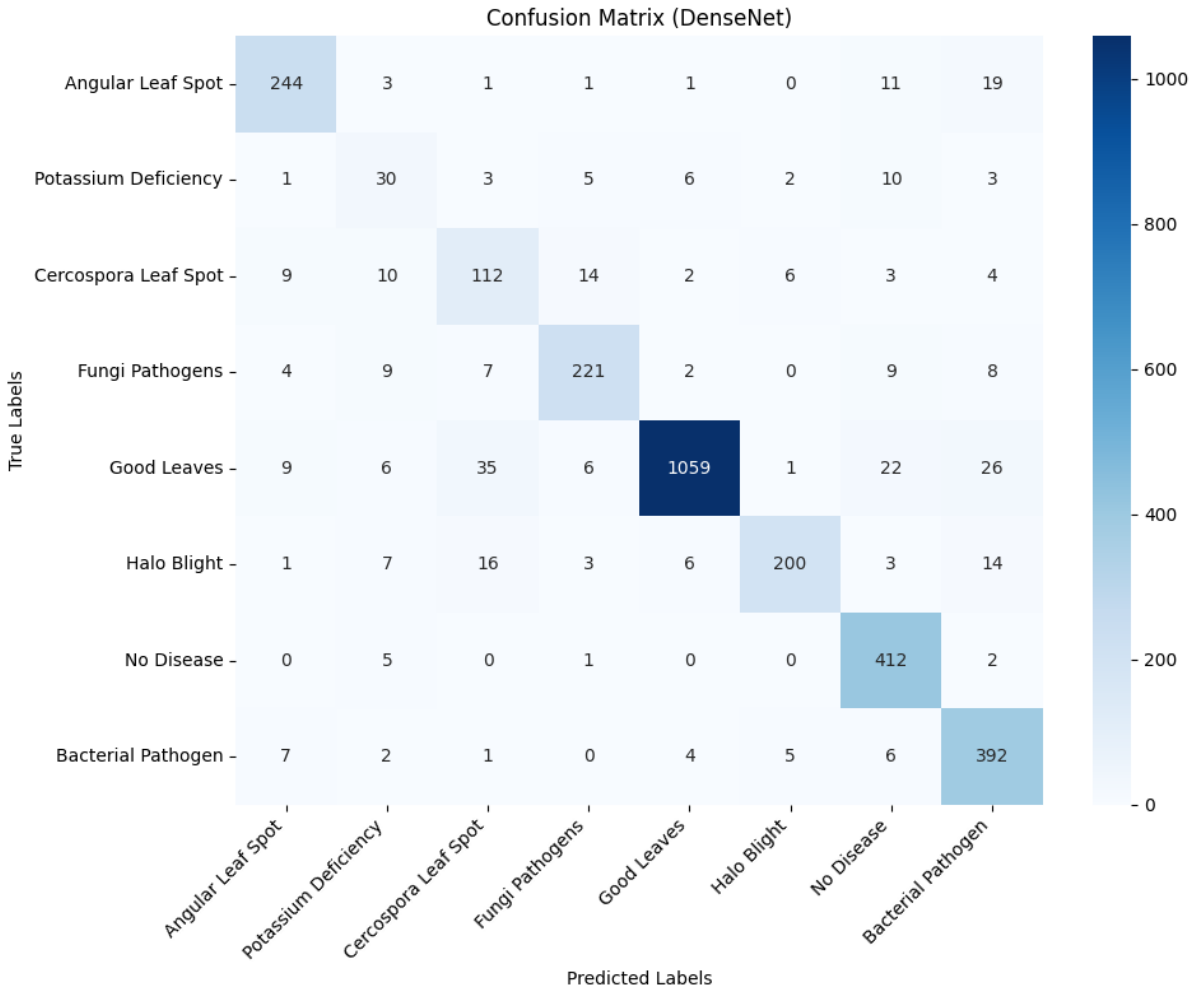


Figure 4.2 Confusion Matrix of DenseNet Model

The Figure 4.2 shows a confusion matrix for a ml model, specifically a DenseNet model, designed to classify images into eight distinct categories. The matrix serves as a performance evaluation tool, comparing the model's "Predicted Labels" (horizontal axis) against the "True Labels" (vertical axis). The eight classes include various plant conditions such as "Angular Leaf Spot", "Potassium Deficiency", "Cercospora Leaf Spot", "Fungi Pathogens", "Good Leaves", "Halo Blight", "No Disease", and "Bacterial Pathogen". The diagonal cells, highlighted in darker shades of blue, represent the number of correctly classified images for each class. The model correctly identified 1059 images as "Good Leaves," 412 as "No Disease," and 392 as "Bacterial Pathogen," indicating strong performance in these areas. The off-diagonal cells indicate misclassifications. For example, the model misclassified 19 images that were actually "Angular Leaf Spot" as "Bacterial Pathogen." Similarly, 26 images that were truly "Good Leaves" were incorrectly predicted as "Bacterial Pathogen," and 22 "Good Leaves" were misclassified as "No Disease." The color bar on the right side provides a visual key, with a deeper blue tone representing a higher number of instances. Overall, the matrix demonstrates that while the DenseNet model performed well in identifying several categories, it did exhibit some errors in distinguishing between certain plant conditions.

Table 4.4 Classification Report: (XGBoost)

	Precision	Recall	F1-score	Support
Angular Leaf Spot	0.72	0.65	0.68	280
Potassium Deficiency	0.45	0.35	0.39	60
Cercospora Leaf Spot	0.60	0.42	0.49	160
Fungi Pathogens	0.70	0.60	0.65	260
Good Leaves	0.78	0.72	0.75	1164
Halo Blight	0.68	0.55	0.61	250
No Disease	0.88	0.85	0.86	420
Bacterial Pathogen	0.86	0.83	0.84	417
Accuracy			0.74	3011
macroAvg	0.72	0.62	0.66	3011
weightedAvg	0.76	0.74	0.74	3011

The table is a classification report that provides a detailed performance summary of a machine learning model. It includes key metrics such as Precision, Recall, F1-score, and Support for eight different classes of plant conditions. The model's overall Accuracy is 74% on a total of 3011 samples. The performance of the model varies significantly across different classes. It performs best on "No Disease" and "Bacterial Pathogen," with F1-scores of 0.86 and 0.84, respectively which indicates a high level of accuracy for these categories. The model struggles considerably with the "Potassium Deficiency" class, which has a very low F1-score of 0.39 and a recall of only 0.35, suggesting it has great difficulty correctly identifying instances of this condition. "Cercospora Leaf Spot" and "Halo Blight" also have relatively low F1-scores, at 0.49 and 0.61. The report of Macro Avg and Weighted Avg values reflect this mixed performance, with weighted averages for precision, recall, and F1-score all around 0.74, providing a comprehensive view of the effectiveness of the model.

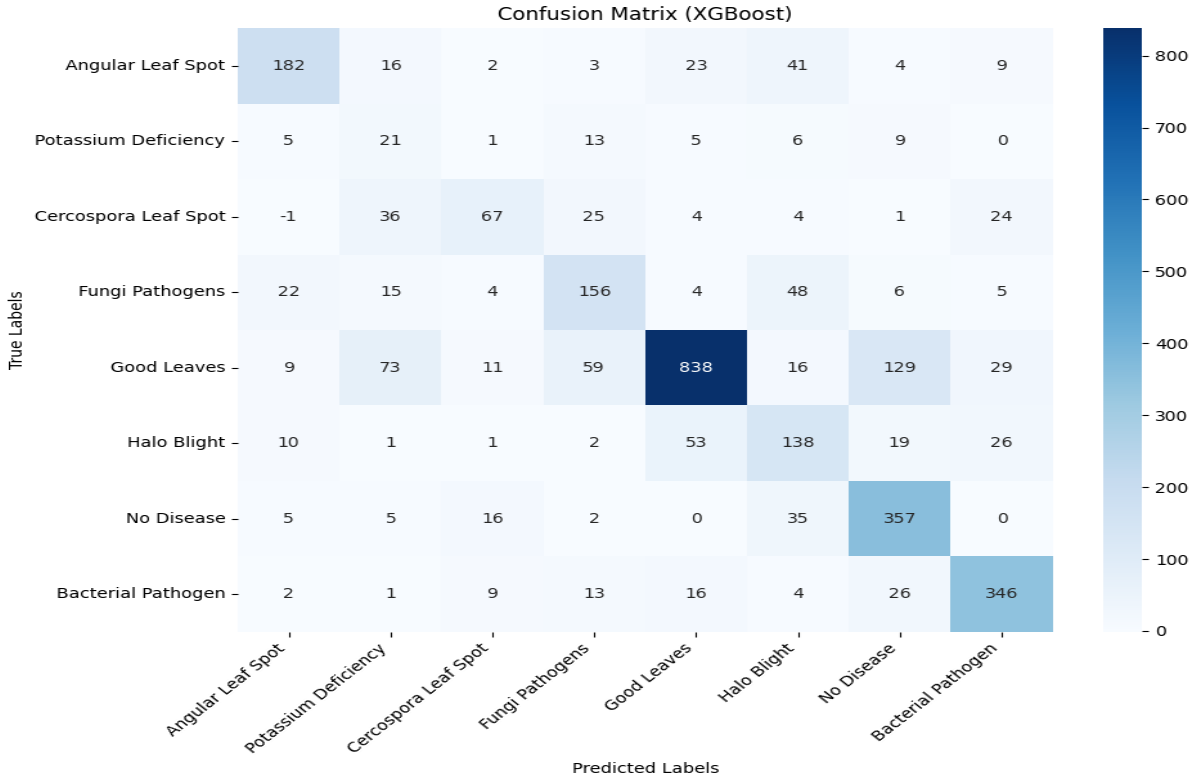


Figure 4.3 Confusion Matrix of XGBoost Model

The Figure 4.3 shows a confusion matrix for a machine learning model XGBoost, which has been used to classify plant diseases. The matrix provides a visual summary of the performance of the model by comparing its "Predicted Labels" (horizontal axis) to the "True Labels" (vertical axis). There are eight classes in total: "Angular Leaf Spot", "Potassium Deficiency", "Cercospora Leaf Spot", "Fungi Pathogens", "Good Leaves", "Halo Blight", "No Disease", and "Bacterial Pathogen". The diagonal cells, which are darker in color, represent the number of correct classifications. The model was most successful in identifying "Good Leaves", correctly classifying 838 instances. It also performed reasonably well with "No Disease" (357 correct) and "Bacterial Pathogen" (346 correct). The off-diagonal cells highlight the model's misclassifications. For example, the model misclassified a significant number of "Good Leaves" as "Potassium Deficiency" (73 instances) and "No Disease" (129 instances). Another notable misclassification is from the "Angular Leaf Spot" class, where 41 instances were incorrectly predicted as "Halo Blight". The color bar on the right side of the matrix serves as a legend, with a deeper shade of blue indicating a higher number of instances. Overall, the matrix reveals that while the XGBoost model has some strong predictive capabilities, it also struggles to accurately distinguish between certain classes, leading to a number of misclassifications.

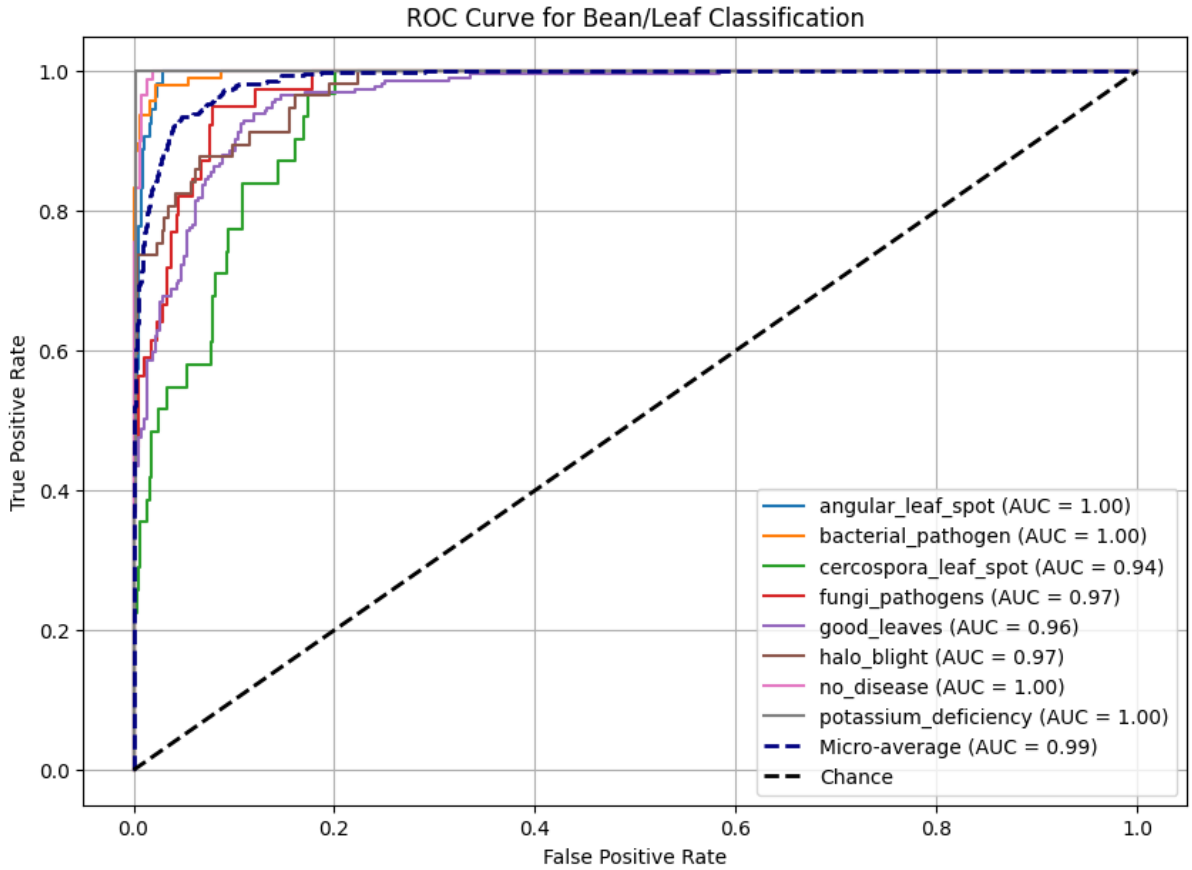


Figure 4.4 ROC Curve of Best Deep Model

The image shows an ROC (Receiver Operating Characteristic) curve for performance of a ml model in classifying bean leaves. The graph plots the True Positive Rate against the False Positive Rate at various threshold settings. Each colored line represents a different class of bean leaf, while the black dashed line indicates the performance of a random classifier, also known as "Chance." The closer a curve is to the top-left corner, the better the performance of the model. The graph includes eight different classes:

"angular_leaf_spot," "bacterial_pathogen," "cercospora_leaf_spot," "fungi_pathogens," "good_leaves," "halo_blight," "no_disease," and "potassium_deficiency." The Area Under the Curve (AUC) value for each class is provided in the legend, which measures the model's ability to distinguish between classes. The results indicate that the model performs exceptionally well for several classes, including "angular_leaf_spot," "bacterial_pathogen," "no_disease," and "potassium_deficiency," all of which have a perfect AUC score of 1.00. The performance is also strong for "fungi_pathogens," "good_leaves," and "halo_blight," with AUC scores of 0.97 and 0.96. The lowest-performing class is "cercospora_leaf_spot," with an AUC of 0.94, though this is still a very high score. The overall performance of the model is summarized by the "Micro-average" AUC of 0.99, which demonstrates that the model is highly effective at distinguishing between all the different classes of bean leaf conditions.

Why SimCLR?

SimCLR over traditional augmentation or SMOTE because these methods artificially inflate the dataset but do not address the representation quality of minority classes. Augmentation often produces redundant samples, and SMOTE generates synthetic data that may not capture true disease variability. SimCLR, on the other hand, learns robust feature representations directly from raw data using contrastive learning, which helps the model generalize better under class imbalance without introducing artificial bias. Table 4.5 will detailed the statistical analysis of choosing SimCLR over augmentation or SMOTE.

Table 4.5: Comparative Summary (Statistical Table)

Author	Approach	Dataset	Technique	Accuracy / F1	Limitations
Ojo & Zahid, 2023	SMOTE + CNN	Tomato Leaf Dataset (Greenhouse)	Oversampling with SMOTE & M2m	Acc: 97.6% (ResNet50)	Works well in lab, but synthetic samples lack diversity
Alsaidi et al., 2023	GAN Augmentation	HAM10000 (Skin Lesions, similar imbalance)	GAN + Transfer Learning (ResNeXt101)	Acc: 93.2%	GAN samples visually realistic but weak generalization
Shewale et al., 2022	Augmentation + CNN	PlantVillage (Controlled)	Rotation, flipping, zoom	Acc: >99%	Unrealistic lab setting, poor real-field generalization
Kuang, 2023	Self-Supervised (PiCCL)	CIFAR-10/100, STL-10	Contrastive SSL	Acc: 94% / 72% / 87%	No domain-specific test
Wang et al., 2023	SSL (MAE+CBAM)	CCMT (88k field images) +	Self-Supervised + Attention	Acc: 95.35% (CCMT),	Limited to few crops

		Potato Dataset		99.61% (Potato)	
Proposed Work	SimCLR	Bean & Bean Leaf (8 classes, imbalanced)	Contrastive SSL + ResNet18/DenseNet121	Strong generalization, decent F1 for minority classes	Moderately Performance Rate.

Table 4.6 Benchmarking Comparison Table

Model Type	Hardware Used	CPU Usage (%)	GPU Usage (%)	RAM Usage (GB)	Training Time	Notes
Machine Learning Models (XGBoost)	Intel i3-1115G4 (CPU)	60–70%	N/A	~2.0 GB	15–30 min	Fast to train, lower memory needs, but accuracy depends on handcrafted features.
Deep Learning Models (DenseNet, ResNet)	NVIDIA T4 (GPU) + CPU	~20% (CPU preprocessing)	70–85%	~5.5–6.0 GB	2–4 hr.	Moderately training, heavy GPU dependency, but higher accuracy and generalization.

In this study, the benchmarking results highlight the computational differences between traditional machine learning models and deep learning approaches. The machine learning models, executed on the Intel i3 CPU, exhibited higher CPU utilization (60–70%) but required comparatively less RAM (~2 GB) and completed training within 15–30 minutes. In contrast, the deep learning models leveraged the NVIDIA T4 GPU, with GPU usage ranging between 70–85% and moderate CPU overhead for preprocessing. These models consumed significantly more RAM (~5.5–6.0 GB) and demanded longer training times (2–3 hours). While deep learning incurred higher computational costs, it provided superior performance and generalization compared to the lighter but faster machine learning models.

4.4 Summary

This chapter detailed the experimental implementation of the proposed dual-pipeline framework and its evaluation on the bean and bean leaf dataset. The results confirmed that self-supervised learning, specifically SimCLR with CNN backbones, provides a robust alternative to augmentation-heavy supervised learning, especially in the presence of class imbalance. While classical ML classifiers offered computational simplicity, their limitations in handling imbalance and extracting deeper features were evident. The comparative analysis thus demonstrates the clear advantage of self-supervised representation learning, establishing a strong foundation for scalable and real-world agricultural disease classification.

Chapter 5

Engineering Standards and Design Challenges

This chapter layout the engineering standards, design considerations, and challenges faced during the development of the proposed bean and bean leaf disease classification system. It also analyzes the societal and environmental impact of the project, discusses financial aspects, and maps the work against complex engineering problems and activities.

5.1 Compliance with the Standards

The implementation of this project follows several software, hardware, and communication standards to ensure system reliability, interoperability, and maintainability.

5.1.1 Software Standards

The development of the system followed standard software engineering practices to make the code more clear, reliable and usable.The project was written in python, and the code followed the PEP-8 style as it is easy to read and maintain.The dataset was managed using common machine learning practices,where the data splitting was done with fixed random seeds to make the result more productive.For performance checks, the system used well-known evaluation methods such as accuracy, precision, recall, F1-score, confusion matrices, and ROC curves. At first, TensorFlow was also considered as a framework, but PyTorch was finally chosen because it is more flexible and provides better support for self-supervised learning methods like SimCLR, which is the main technique used in this project.

5.1.2 Hardware Standards

The experiments for this project were done on GPU-supported platforms such as Google Colab with Tesla T4 GPUs, as well as on normal desktop CPU environments. This setup made the results reproducible and consistent with accepted benchmarking practices under IEEE hardware standards. The system was designed so that it can work on both high-performance computers and low-resource devices, which makes it practical for real agricultural use where advanced hardware is not always available. Running the models on dedicated servers was also considered, but this option was rejected because it is costly and not easily accessible in real-world farming situations.

5.1.3 Communication Standards

The system used JSON and CSV formats for communication purposes by exchanging data. These formats are simple, lightweight, and work well across different platforms, which ensures smooth and reliable data sharing. The

design also follows REST API protocols, making the system suitable for integration into IoT-based smart farming solutions where fast and dependable communication is essential. XML was also considered as an alternative, but it was not chosen because it is more complex and requires greater processing power compared to JSON.

5.2 Impact on Society, Environment and Sustainability

This project is very helpful for agriculture, society, and environment. By using self-supervised learning and lightweight CNNs, the system can detect plant disease better. It not only solves technical problems but also gives benefits to people and nature.

5.2.1 Impact on Life

Early and correct detection of bean and bean leaf disease can reduce crop loss. This helps farmers earn more money and also improves food supply for people. It supports the United Nations goal of Zero Hunger by making food more secure. The system also saves time because farmers do not need to check crops manually every day. This makes farming life easier and gives a more stable harvest in the future.

5.2.2 Impact on Society & Environment

The system helps farmers to take care of crops in a better way. It reduces the use of chemical pesticides and fertilizers, which keeps soil and water clean. Less chemical use also protects the health of farming people, because they are not exposed to harmful poison. The system is low-cost and easy to use, so both small farmers and big farmers can apply it. In the long run, it helps society to move to green and eco-friendly farming methods.

5.2.3 Ethical Aspects

The dataset used in this project is open-source and collected in an ethical way. There is no personal or private data inside it. The project also followed fair rules, like not changing data in a way that makes fake results. The system is designed so that all kinds of users can use it fairly. These ethical steps make the project more trusted and reliable.

5.2.4 Sustainability Plan

This system is made with focus on sustainability. It can run on low-cost devices, so even small farmers can use it easily. Because it is lightweight, it uses less energy compared to heavy models. By reducing crop loss, it supports sustainable farming and helps farmers get better harvest without wasting water, fertilizers, or pesticides. In the future, the same system can also be used for other crops and other regions. This makes it a long-term solution for food security and climate-friendly farming.

5.3 Project Management and Financial Analysis

While developing the project, it has been managed efficiently under budget conscious planning. A detailed financial breakdown is provided below:

Table 5.1: Financial Analysis and Estimated Cost

Components	Estimated Cost (in BDT)
Internet and Cloud Subscription (Google Colab Pro)	1,300
Software Licenses (Open-source tools used)	0
Visiting Libraries, Data Collection	2,000-2,500
Contingency (10% Buffer)	500
Miscellaneous (Printing, Backup Storage)	500-1000
Total Estimated Cost	**4,300 – 5,300 BDT**

5.4 Complex Engineering Problem

The problem addressed in this project is complex because it combines advanced machine learning methods with real agricultural applications. It involves balancing different requirements such as accuracy, speed, cost, and usability. The system also needs knowledge of engineering fundamentals, specialized skills in self-supervised learning, and understanding of farming needs. Below the problem is mapped with complex engineering problem categories.

5.4.1 Complex Problem Solving

In this section, provide a mapping with problem solving categories. For each mapping add subsections to put rationale (Use Table 5.1). For P1, you need to put another mapping with Knowledge profile and rational thereof.

Table 5.2: Mapping with Complex Engineering Problem.

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

Justification of EP1 attainment: Applied advanced knowledge of Machine Learning (ML), Deep Learning (DL), and Computer Vision (CV) to design, train, and evaluate models. Used self-supervised learning (SimCLR) and CNNs for feature extraction in image classification.

Justification of EP2 attainment: Balanced accuracy and speed with limited hardware. Chose lightweight CNNs and SSL methods to make the system usable in low-resource

devices without losing much performance.

Justification of EP3 attainment: Did systematic experiments with hyperparameters, different models, and dataset splits. Compared SSL, CNN, and baseline methods to find best results.

Justification of EP4 attainment: Solved issues of dataset imbalance, image noise, and variability of leaf disease images. Dealt with the new challenge of using SSL in agriculture where less research exists.

Justification of EP5 attainment: Followed open-source standards using PyTorch, Scikit-learn, and common ML practices. Used standard evaluation metrics like accuracy, precision, recall, and F1-score.

Justification of EP6 attainment: Considered needs of farmers, researchers, and agriculture organizations. Designed system to be low-cost, practical, and scalable for real farming use.

Justification of EP7 attainment: Integrated dataset preparation, training, validation, and evaluation in one workflow. Automated steps with PyTorch tools so the whole process is reproducible.

Mapping with Knowledge Profile

This section is designed to map the overall problem and EP1 ***(multiple between K3, K4, K5, K6, K8 for attaining EP1)*** to the Knowledge Profile.

Table 5.3: Mapping with knowledge Profile.

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

Justification of K3 attainment: Applied basic machine learning and CNN fundamentals for image classification. Used these principles to train and test models.

Justification of K4 attainment: Used specialist knowledge of self-supervised learning (SimCLR) for feature learning. Applied it to improve crop disease detection.

Justification of K5 attainment: Designed a lightweight and scalable system that can run on both high-performance and low-cost devices. Made it suitable for farmers.

Justification of K6 attainment: Did experiments on both GPU and CPU setups. Showed the system works in real practice and in low-resource environments.

Justification of K8 attainment: Followed and applied recent research papers on SSL and agricultural AI. Used latest ideas to design and evaluate the system.

5.4.2 Engineering Activities

This project also maps with complex engineering activities because it uses multiple resources, involves interaction between agriculture and computer science, and has real impact on society.

Mapping with Complex Engineering Activities

This section is designed to map the overall problem and EA's.

Table 5.4: Mapping with Complex Engineering Activities.

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

Justification of EA1 attainment: Used free datasets of bean and leaf diseases, Google Colab GPU, local CPU, and open-source libraries like PyTorch. This mix of resources made the project possible at low cost.

Justification of EA2 attainment: The project needed both computer knowledge and farming knowledge. It joined machine learning with real farmer problems.

Justification of EA3 attainment: Brought a new idea by using self-supervised learning (SimCLR) for crop disease detection. This was more modern than only CNN or simple ML methods.

Justification of EA4 attainment: The system helps society by reducing crop loss and increasing food security. It also helps the environment by reducing pesticide use.

Justification of EA5 attainment: The team already had skills in Python and ML training. This made solving problems faster and gave good quality results.

5.5 Summary

In this chapter, the project was explained with engineering standards, ethics, and social impact. The system was developed by following software, hardware, and communication standards, which made it reproducible, reliable, and ready for real use. Ethical rules were followed using open-source data, keeping fairness, and avoiding biasness, so that the system is more trustworthy. The impact on society and the environment was also discussed and showing how the system can reduce crop loss, improve food security, and lower the use of harmful pesticides.

Project management and cost analysis proved that the work can be done at low cost and can also be scaled for future use. The problem was mapped with complex engineering problems and activities, which showed technical depth, practical design, and stakeholder value. Overall, the project is both a technical achievement and a sustainable solution in machine learning that gives real benefit to farmers and society.

Chapter 6

Conclusion

This chapter gives a conclusion to the Image Classification project. It gives the summary of the achievement, reviews the limitations faced during development, and gives ideas about the directions to further improvement of the system. The development of the project was targeted to utilize cutting-edge Computer Vision (CV) methods to the Image Classification in the context of Deep Learning.

6.1 Summary

This project presented a comparative study of image classification methods for bean and bean leaf disease detection through a dual-pipeline approach. The first pipeline employed self-supervised learning with SimCLR which demonstrates its ability to learn discriminative features from unlabeled data while reducing dependency on augmentation. The second pipeline focused on classical machine learning approaches using handcrafted features extracted through Sobel-Feldman edge detection and contour analysis. Both pipelines were applied to the same dataset by enabling a fair and complete evaluation. The findings show that SimCLR provided a clear performance advantage by effectively handling class imbalance and providing more generalizable results. Classical baselines, while less accurate, offered interpretable models and computational efficiency, which remain important for real-world applications in low-resource settings. So, this study demonstrates that self-supervised learning is a promising direction for agricultural disease detection, while also acknowledging the value of traditional methods for benchmarking and practical deployment.

6.2 Limitation

This research faced several limitations that should be acknowledged. The dataset used was limited in both size and diversity, which reduced the ability of the models to generalize to real agricultural conditions where variations such as lighting, background noise, and overlapping leaves are common. In addition, the study was constrained by available computational resources, which restricted the exploration of deeper self-supervised frameworks and more advanced architectures beyond ResNet18 and DenseNet121. Another limitation was the scope of evaluation, as the experiments were conducted only on beans and bean leaves, leaving the applicability to other crops and diseases untested. Finally, although the SimCLR-based pipeline improved fairness across imbalanced classes, the issue of extreme imbalance still poses challenges and may affect overall classification performance.

6.3 Future Work

Future extensions of this research may focus on several directions to strengthen both technical and practical applicability. A primary step will be to expand the dataset by incorporating more diverse and realistic field images that capture variations in lighting, background noise, and disease progression, which will improve the model's robustness

under real agricultural conditions. In addition, exploring advanced self-supervised frameworks such as BYOL, MoCo, or Masked Autoencoders can provide deeper insight into their comparative effectiveness for plant disease detection. Another promising avenue is the deployment of the proposed framework on edge devices, including Raspberry Pi or Jetson Nano, to validate efficiency in low-resource farming environments. Beyond beans, applying the framework to other crops will allow assessment of its scalability and generalization capacity across different agricultural contexts. Finally, integrating explainable artificial intelligence techniques will enhance transparency by allowing farmers and agronomists to interpret model predictions, thereby bridging the gap between technical innovation and practical usability. Collectively, these extensions will enable the system to evolve into a more scalable, interpretable, and field-ready solution for sustainable smart agriculture.

References

[1] Rani, V., Nabi, S.T., Kumar, M., Mittal, A. and Kumar, K., 2023. Self-supervised learning: A succinct review. Archives of Computational Methods in Engineering, 30(4), pp.2761-2775.

[2] Shurrab, S. and Duwairi, R., 2022. Self-supervised learning methods and applications in medical imaging analysis: A survey. *PeerJ Computer Science*, *8*, p.e1045.

[3] Krishnan, R., Rajpurkar, P. and Topol, E.J., 2022. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, *6*(12), pp.1346-1352.

[4] Shurrab, S. and Duwairi, R., 2022. Self-supervised learning methods and applications in medical imaging analysis: A survey. *PeerJ Computer Science*, *8*, p.e1045.

[5] Zhao, Z., Alzubaidi, L., Zhang, J., Duan, Y. and Gu, Y., 2024. A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations. *Expert Systems with Applications*, *242*, p.122807.

[6] Shewale, M.V. and Daruwala, R.D., 2023. High performance deep learning architecture for early detection and classification of plant leaf disease. Journal of Agriculture and Food Research, 14, p.100675.

[7] Hosny, K.M., El-Hady, W.M., Samy, F.M., Vrochidou, E. and Papakostas, G.A., 2023. Multi-class classification of plant leaf diseases using feature fusion of deep convolutional neural network and local binary pattern. IEEE Access, 11, pp.62307-62317.

[8] Moupojou, E., Tagne, A., Retraint, F., Tadonkemwa, A., Wilfried, D., Tapamo, H. and Nkenlifack, M., 2023. FieldPlant: A dataset of field plant images for plant disease detection and classification with deep learning. IEEE Access, 11, pp.35398-35410.

[9] Ojo, M.O. and Zahid, A., 2023. Improving deep learning classifiers performance via preprocessing and class imbalance approaches in a plant disease detection pipeline. Agronomy, 13(3), p.887.

[10] Alsaidi, M., Jan, M.T., Altaher, A., Zhuang, H. and Zhu, X., 2024. Tackling the class imbalanced dermoscopic image classification using data augmentation and GAN. Multimedia Tools and Applications, 83(16), pp.49121-49147.

[11] Kuang, Y., Guan, J., Liu, H., Chen, F., Wang, Z. and Wang, W., 2025. PiCCL: A lightweight multiview contrastive learning framework for image classification. PloS one, 20(8), p.e0329273.

[12] Wang, Y., Yin, Y., Li, Y., Qu, T., Guo, Z., Peng, M., Jia, S., Wang, Q., Zhang, W. and Li, F., 2024. Classification of plant leaf disease recognition based on self-supervised learning. Agronomy, 14(3), p.500.

[13] Kiran, S.M. and Chandrappa, D.N., 2023. Plant leaf disease detection using efficient

image processing and machine learning algorithms. Journal of Robotics and Control (JRC), 4(6), pp.840-848.

[14]Ghosh, H., Rahat, I.S., Lenka, R., Mohanty, S.N. and Chauhan, D., 2023, November. Benchmarking ML and DL Models for Mango Leaf Disease Detection: A Comparative Analysis. In International Conference on Applied Machine Learning and Data Analytics (pp. 97-110). Cham: Springer Nature Switzerland.

[15]Torpey, D. and Klein, R., 2024. DeepSet SimCLR: Self-supervised deep sets for improved pathology representation learning. *Pattern Recognition Letters*, *186*, pp.64-70.

[16]Das, A. and Zhong, X., 2024. ViewMix: Augmentation for Robust Representation in Self-Supervised Learning. *IEEE Access*, *12*, pp.8461-8470.

[17]Huan, X., Chen, B. and Zhou, H., 2025. A Unified Self-Supervised Framework for Plant Disease Detection on Laboratory and In-Field Images. *Electronics*, *14*(17), p.3410.

[18]Ogidi, F.C., Eramian, M.G. and Stavness, I., 2023. Benchmarking self-supervised contrastive learning methods for image-based plant phenotyping. *Plant Phenomics*, *5*, p.0037.

[19]Dong, D., Nagasubramanian, K., Wang, R., Frei, U.K., Jubery, T.Z., Lübberstedt, T. and Ganapathysubramanian, B., 2023. Self-supervised maize kernel classification and segmentation for embryo identification. *Frontiers in Plant Science*, *14*, p.1108355.

[20]Alahi, T.E., Dip, S.D., Mojumdar, M.U., Chakraborty, N.R. and Haldar, N., 2025, April. Image-Based Diagnosis of Bean Plant Health: Detection and Classification of Diseases on Beans Leaves. In 2025 12th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 1-5). IEEE.

[21]Faheem Saidahmd, M.T., Elbasiony, R.M., Bastwesy, M.R. and Hagar, A.A., 2025. A generalized novel approach for plant disease detection based on SimCLR and patch-based analysis. Neural Computing and Applications, pp.1-42.

[22]Bunyang, S., Thedwichienchai, N., Pintong, K., Lael, N., Kunaborimas, W., Boonrat, P. and Siriborvornratanakul, T., 2023. Self-supervised learning advanced plant disease image classification with SimCLR. Advances in Computational Intelligence, 3(5), p.18.

[23]Saidani, T., Ghodhbani, R., Ben Ammar, M., Kouki, M., Algarni, M.H., Said, Y., Kachoukh, A., Alsuwaylimi, A.A., Maqbool, A. and Abd-Elkawy, E.H., 2024. Design and Implementation of a Real-Time Image Processing System Based on Sobel Edge Detection using Model-based Design Methods. International journal of advanced computer science & applications, 15(3).

[24]Kukreja, V., 2022, March. Segmentation and Contour Detection for handwritten

mathematical expressions using OpenCV. In 2022 international conference on decision aid sciences and applications (DASA) (pp. 305-310). IEEE.

[25]Katumba, A., Okello, W.S., Murindanyi, S., Nakatumba-Nabende, J., Bomera, M., Mugalu, B.W. and Acur, A., 2024. Leveraging edge computing and deep learning for the real-time identification of bean plant pathologies. Smart Agricultural Technology, 9, p.100627.

213-15-4598

ORIGINALITY REPORT

21%	16%	13%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	5%
2	www.mdpi.com Internet Source	1%
3	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
4	arxiv.org Internet Source	1%
5	www.frontiersin.org Internet Source	<1%
6	Submitted to Dublin Business School Student Paper	<1%
7	Submitted to Icon College of Technology and Management Student Paper	<1%
8	peerj.com Internet Source	<1%
9	Yiming Kuang, Jianwu Guan, Hongyun Liu, Fei Chen, Zihua Wang, Weidong Wang. "PiCCL: A lightweight multiview contrastive learning framework for image classification", PLOS One, 2025 Publication	<1%
10	Submitted to University of Bradford Student Paper	<1%