

Watermelon Leaf Disease Detection Using Machine Learning and Deep Learning Based Hybrid Approach

By
Razone Parvej Rabbi
ID : 211-15-3999

FINAL YEAR DESIGN PROJECT REPORT

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

Supervised by

Ms. Amatul Bushra Akhi
Assistant Professor

Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

Umme Ayman
Lecturer

Department of Computer Science and Engineering
Daffodil International University



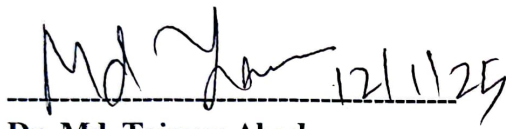
**DAFFODIL INTERNATIONAL
UNIVERSITY**
Dhaka, Bangladesh

January 12, 2025

APPROVAL

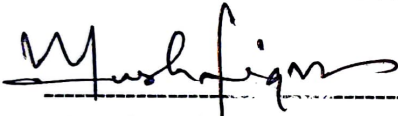
This Project titled “Watermelon Leaf Disease Detection Using Machine Learning and Deep Learning Based Hybrid Approach”, submitted by **Razone Parvej Rabbi**, ID No: 211-15-3999 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for 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 12th January, 2025.

BOARD OF EXAMINERS



Dr. Md. Taimur Ahad
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



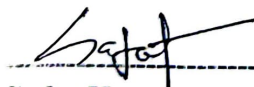
Mushfiqur Rahman
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Mr. Rahmatul Kabir Rasel Sarker
Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



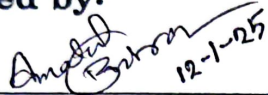
Sadat Hasan
Data Scientist (Senior Principal Officer)
Risk Management Division
BRAC Bank

External Examiner

DECLARATION

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

Supervised by:


12-1-25

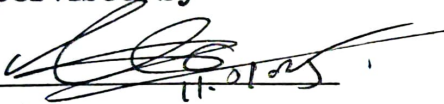
Ms. Amatul Bushra Akhi

Assistant Professor

Department of Computer Science and Engineering

Daffodil International University

Co-Supervised by:


11-01-25

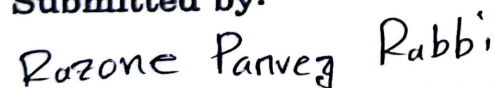
Umme Ayman

Lecturer

Department of Computer Science and Engineering

Daffodil International University

Submitted by:



Rzone Parvej Rabbi

Student ID: 211-15-3999

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 **Ms. Amatul Bushra Akhi, 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 **Machine Learning** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

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

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

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

ABSTRACT

Watermelon diseases significantly affect agricultural productivity, leading to economic losses and reduced crop quality. Traditional manual inspection methods are time-consuming, labor-intensive, and susceptible to human error. This research explores various Machine Learning (ML) and Deep Learning (DL) approaches to classify watermelon leaf diseases, with an emphasis on identifying the best-performing models for integration into a hybrid classification system. A dataset of approximately 5000 images, including healthy and diseased leaf samples, was sourced from Kaggle and preprocessed to ensure robust training.

Initially, ML algorithms such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, and Random Forest were tested, yielding accuracies ranging from 59% to 88%. DL architectures, including baseline CNN, ResNet50, MobileNetV2, DenseNet121, and InceptionV3, were subsequently evaluated, with ResNet50 achieving the highest accuracy of 99.75%, closely followed by MobileNetV2 with 99%. Based on these findings, a hybrid model was constructed by combining SVM (for classification) and a pre-trained ResNet50 (for feature extraction), achieving an accuracy of 99.80%.

This study demonstrates how artificial intelligence can be used practically to advance precision farming by promoting sustainable agricultural practices. By integrating ML and DL techniques into a hybrid model, this research contributes a significant step toward more accurate and impactful solutions for watermelon leaf disease classification, supporting sustainable agriculture and global food security.

Table of Contents

Approval	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Introduction.....	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Methodology	3
1.5 Project Outcome.....	5
1.6 Organization of the Report.....	5
2 Background	6
2.1 Introduction.....	6
2.2 Literature Review.....	6
2.2.1 Similar Applications.....	7
2.2.2 Related Research	8
2.3 Gap Analysis	8
2.4 Summary	9
3 Research Methodology	10
3.1 Methodology/Requirement Analysis & Design Specification	10
3.1.1 Overview	10
3.1.2 Proposed Methodology.....	10
3.2 Detailed Methodology.....	13
3.3 Project Plan.....	22

3.4	Task Allocation.....	23
3.5	Summary.....	24
4	Implementation and Results	25
4.1	Environment Setup	25
4.2	Comparative Analysis	26
4.3	Results and Discussion.....	30
4.4	Summary	31
5	Engineering Standards and Design Challenges	32
5.1	Compliance with the Standards	32
5.1.1	Software Standards	32
5.1.2	Hardware Standards.....	33
5.1.3	Communication Standards.....	33
5.2	Impact on Society, Environment and Sustainability	33
5.2.1	Impact on Life	33
5.2.2	Impact on Society & Environment.....	34
5.2.3	Ethical Aspects	34
5.2.4	Sustainability Plan.....	34
5.3	Project Management and Financial Analysis.....	35
5.4	Complex Engineering Problem.....	35
5.4.1	Complex Problem Solving.....	35
5.4.2	Engineering Activities	39
5.5	Summary	40
6	Conclusion	41
6.1	Summary	41
6.2	Limitation	41
6.3	Future Work.....	42
	References	43

List of Figures

Figure 3.1.2.1: Methodology Diagram	12
Figure 3.2.1 : Background removal example	13
Figure 3.2.2 : Image enhancement example	14
Figure 3.2.3 : Results of applied models.....	14
Figure 4.2.1 : SVM Confusion Matrix	29
Figure 4.2.2 : Resnet50 Confusion Matrix	29
Figure 4.2.3 : Hybrid Model Confusion Matrix.....	29
Figure 4.2.4 : Hybrid Model PCA of Test Features	30
Figure 4.3.1 : Best performing models comparison.....	31

List of Tables

2.2.1: Summary of Literature Reviewed.	7
2.3.1: Gap Analysis Table.....	9
3.2.1: Results of applied models.....	20
5.4.1.1: Mapping with complex problem solving.....	37
5.4.1.2: EP1 Knowledge Profiles (K).....	37
5.4.1.3: EP2 Knowledge Profiles (K).....	38
5.4.1.4 : EP3 Knowledge Profiles (K).....	38
5.4.1.5: EP4 Knowledge Profiles (K).....	38
5.4.1.6: EP7 Knowledge Profiles (K).....	39
5.4.1.7: EA Standards Identification.	40

Chapter 1

Introduction

1.1 Introduction

The rapid expansion of global agriculture industries and their need for disease-free, high-quality produce has placed increased importance on improving disease detection techniques for various crops. Among these, watermelon cultivation stands as a significant contributor to global fruit markets, yet it remains vulnerable to numerous plant diseases. Traditional methods with manual inspection often cost a lot of time, is labor-intensive, and prone to human error. To solve these problems, this project focuses on the development and evaluation of a watermelon disease detection system using image recognition and classification techniques. By applying both Deep Learning (DL) and Machine Learning (ML) methodologies. The goal is to identify and classify watermelon leaf and fruit diseases accurately, thus enabling early intervention and improving overall crop yield.

1.2 Motivation

The motivation behind this research arises from the need to overcome computational and practical challenges in accurately identifying watermelon leaf diseases. Traditional manual methods for disease detection are labor-intensive, time-consuming, and prone to human error, often leading to delayed or incorrect diagnoses. This inefficiency can result in significant crop losses and reduced agricultural productivity, emphasizing the necessity of precise and efficient classification methods.

With the advent of advanced Machine Learning (ML) and Deep Learning (DL) techniques, it is now possible to leverage computational power to address

these challenges. This project systematically explores a range of ML algorithms SVM, KNN, Decision Trees, and Random Forest and state-of-the-art DL architectures, including ResNet50, MobileNetV2, DenseNet121, InceptionV3, and a baseline CNN. These methods were applied to a large, augmented dataset of approximately 5000 images representing both healthy and diseased leaves to evaluate their performance and accuracy.

The ultimate motivation lies in combining the strengths of ML and DL approaches to achieve unprecedented accuracy through a hybrid model. Specifically, the integration of SVM for classification and ResNet50 for feature extraction not only showcases the synergy between these techniques but also highlights their potential to deliver results with almost near perfect accuracy. Such advancements demonstrate how artificial intelligence can play a transformative role in precision agriculture, enabling effective disease management while promoting sustainability.

This research also serves as an opportunity to expand the researcher's expertise in machine learning, deep learning, and hybrid modeling techniques. Beyond the academic significance, the project holds immense practical value for farmers, agricultural researchers, and technology developers by contributing to scalable and high-precision solutions for watermelon disease management.

1.3 Objectives

The main objectives of this research are:

1. To collect and preprocess a comprehensive dataset of approximately 5000 healthy and diseased watermelon leaf images from sources like Kaggle, ensuring proper preprocessing for robust training and evaluation by removing unnecessary background and enhancing image details.
2. To systematically evaluate and compare the performance of various ML algorithms (SVM, KNN, Decision Tree, Random Forest) and DL

architectures (CNN, ResNet50, MobileNetV2, DenseNet121, InceptionV3) in classifying watermelon leaf diseases.

3. To identify the best-performing ML and DL models and construct a hybrid model, leveraging SVM for classification and ResNet50 (pre-trained) for feature extraction, achieving maximum classification accuracy.
4. To demonstrate the practicality of hybrid modeling techniques in enhancing the accuracy and scalability of watermelon disease classification.

1.4 Methodology

The methodology of this research involves a series of structured and systematic steps to achieve the objectives:

1. Dataset Preparation:

A dataset containing approximately 5000 images of healthy and diseased watermelon leaves (after augmentation) was curated from Kaggle. The dataset was categorized into three distinct classes: Healthy, Downy Mildew, and Mosaic Virus. The images underwent preprocessing to remove unnecessary background of the image and enhancing the necessary objects details by increasing the images contrast and sharpness while reducing minimum noise to enhance the dataset's robustness for training machine learning (ML) and deep learning (DL) models.

2. Implementation of Machine Learning Models:

Several traditional ML algorithms, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, and Random Forest, were trained and evaluated. These models were assessed based on classification accuracy and other performance metrics such as precision, recall, and F1-score.

3. Implementation of Deep Learning Architectures:

Advanced DL architectures were employed for feature extraction and classification. These included a baseline CNN, ResNet50, MobileNetV2, DenseNet121, and InceptionV3. The models were implemented using Python and trained on Google Colab's GPU/TPU resources to ensure sufficient computational efficiency. Special emphasis was placed on evaluating ResNet50 due to its superior performance.

4. Hybrid Model Development:

After evaluating individual performances, a hybrid model was developed by combining SVM (for classification) and the pre-trained ResNet50 (for feature extraction). This hybrid model was designed to leverage the strengths of both approaches, achieving a near perfect accuracy in watermelon leaf disease classification.

5. Performance Evaluation:

All models, including the hybrid model, were rigorously tested using validation metrics such as accuracy, precision, recall, and F1-score. Additional manual testing on individual images from the dataset ensured robustness.

6. Insights and Practical Applications:

This research emphasizes the computational accuracy of classification methods. By combining ML and DL approaches into a high-precision hybrid model, this study demonstrates the potential of artificial intelligence in disease classification.

The structured methodology demonstrates a blend of traditional ML techniques, state-of-the-art DL architectures, and an innovative hybrid approach, ensuring comprehensive and precise results while paving the way for future advancements in agricultural AI systems.

1.5 Project Outcome

The primary outcome of this research is the development of a highly accurate hybrid model for watermelon leaf disease detection. By systematically evaluating both traditional machine learning techniques, such as SVM, KNN, Decision Tree, and Random Forest, and advanced deep learning architectures like CNN, ResNet50, MobileNetV2, DenseNet121, and InceptionV3, the study identified ResNet50 as the top performer with 99.75% accuracy. The key contribution lies in combining SVM for classification and ResNet50 (pre-trained) for feature extraction to create a hybrid model. This integration resulted in a model that achieved 99.80% accuracy, a slightly better accuracy demonstrating the effectiveness of blending machine learning and deep learning methods for more precise and reliable disease detection, highlighting the potential for such hybrid approaches to address challenges in agricultural disease management.

1.6 Organization of the Report

This report is structured into several chapters for clarity and coherence. After this introductory chapter, Chapter 2 provides a comprehensive review of already existing literatures on plant disease detection and related methodologies. Chapter 3 outlines the detailed methodology, including data preprocessing steps and model selection criteria. Chapter 4 describes the experimental setup, training procedures, computational environment, and performance evaluation metrics. Chapter 5 discusses about the software and hardware standards, effect on society and environment, project management and financial analysis, Complex Engineering problems and Engineering Activities faced for this research. Finally, Chapter 6 concludes the report by summarizing the key findings, acknowledging the limitations, and suggesting potential avenues for future work.

Chapter 2

Background

2.1 Introduction

Accurately identifying and classifying plant diseases is essential to maintain crop health and yield in the context of precision agriculture. Watermelon (*Citrullus lanatus*), a fruit that is grown all over the world, is susceptible to a number of illnesses that includes Mosaic Virus, Anthracnose, Downy Mildew and many other diseases. Identifying these diseases promptly can significantly reduce crop losses, enhance yields, and ensure economic sustainability. Recent advancements in machine learning (ML), image recognition and deep learning (DL) offer automated, efficient, and scalable approaches to disease detection. These technological solutions enable the processing of large datasets, extraction of meaningful features, and real-time prediction capabilities facilitating timely and informed decisions in agricultural management.

2.2 Literature Review

The literature reveals a progression from traditional rule-based and statistical models to more sophisticated ML and DL approaches. Early studies relied on expert systems incorporating domain knowledge, while later research focused on extracting features from leaf images using classical ML techniques. More recent studies leverage convolutional neural networks (CNNs), transfer learning, and hybrid models to achieve higher accuracy and robustness in classifying watermelon diseases. In addition, emerging methods consider data augmentation, hyperspectral imaging, and the incorporation of remote sensing data to enhance the reliability and applicability of detection models.

To provide an overview, Table 2.1 presents a summary of the selected literature,

detailing methodologies, key findings, and reported accuracy or performance metrics.

Table 2.2.1: Summary of Literature Reviewed

Ref.	Methodology	Key Findings	Accuracy/Performance
[1]	Rule-Based Expert System	Accurate diagnosis for Downy Mildew, Anthracnose, Powdery Mildew using domain knowledge	Not specified
[2]	Disease Overview	Highlights economic impact & need for early detection in watermelon	Informative background
[3]	Neural Network (RGB Features)	Classified Anthracnose & Downy Mildew using color indices	75.9% accuracy
[4]	Image Recognition (Multispectral)	Addressed background noise and variability in plant images	Enhanced robustness
[5]	Transfer Learning (MobileNet)	Improved classification with limited datasets (non-watermelon defects)	97.69% accuracy
[6]	ANN (Pixel-Based Features)	Distinguished crops/weeds; potential for adaptation in disease detection	Up to 100% (corn)
[7]	CNN (Pre-trained Models)	High accuracy in general plant health classification	99.58% accuracy
[8]	Survey on Disease Management	Identified top research priorities in watermelon disease management	Not applicable
[9]	Genetic Resources Analysis	Emphasized genetic diversity for disease resistance in Citrullus	Not applicable
[10]	Disease Impact Analysis (Powdery Mildew)	Emphasized need for resistant cultivars in US regions	Not specified
[11]	Optimized CNN (VGG-16)	Mitigated overfitting to improve recall & TPR in watermelon disease stages	Recall: 0.9394; TPR: 98%
[12]	Stacked RNN (Melons)	High accuracy & efficiency in disease classification, similar domain	High accuracy
[13]	CNN+SVM Hybrid	Classified eight watermelon leaf diseases effectively, handling imbalance	Precision: 70.26–76%; Recall: 75.56–98%
[14]	Remote Sensing + ML (Downy Mildew)	Classified severity stages using hyperspectral data	Up to 91% accuracy
[15]	Review (ML/DL Techniques)	YOLOv5 & ResNet50/MobileNetv2 effective for object detection & classification	High accuracy, balanced performance
[16]	Hierarchical Framework (Deep Feature Fusion + SVM)	Enhanced fruit leaf disease classification using feature selection	Accuracy: 97.8%

2.2.1 Similar Applications

Several works focus directly on watermelon disease detection and management. A rule-based expert system effectively diagnosed common

watermelon diseases [1], while comprehensive disease overviews emphasized the importance of early detection [2]. Neural network approaches utilizing RGB features targeted Anthracnose and Downy Mildew [3]. Surveys identified research priorities for disease management [8], and studies highlighted the emergence and economic implications of diseases like Powdery Mildew [10]. Advanced DL models, such as an optimized VGG-16-based CNN, addressed overfitting issues to improve classification performance [11]. Hybrid CNN+SVM frameworks effectively handled class imbalance in multiple watermelon leaf diseases [13], and hyperspectral imaging combined with ML achieved up to 91% accuracy in Downy Mildew severity classification [14].

2.2.2 Related Research

Beyond watermelon-specific applications, various ML and DL techniques have been explored for general plant disease detection. Multispectral and hyperspectral imaging methods improved robustness in noisy environments [4]. Transfer learning approaches, such as using MobileNet for defect detection, demonstrated high accuracy even with limited data [5]. Studies employing ANN and CNN architectures achieved remarkable results in classifying crops, weeds, and plant health states [6], [7]. Genetic diversity analyses offered insights into breeding disease-resistant cultivars [9], and RNN-based models proved effective for other melon species [12]. Comprehensive reviews identified state-of-the-art object detection and classification models (e.g., YOLOv5, ResNet50, MobileNetv2) and hierarchical frameworks incorporating deep feature fusion and selection for improved accuracy [15], [16].

2.3 Gap Analysis

This gap analysis table highlights the key areas where current research falls short and identifies potential opportunities for advancement in watermelon disease detection systems.

While existing research demonstrates significant progress, gaps remain in the practical and scalable deployment of watermelon disease detection systems.

Also finding the best possible method to classify watermelon disease is the key target of this research. Key challenges includes the limited availability of high-quality, annotated datasets; varying environmental conditions that introduce noise and reduce model generalizability; and the difficulty of achieving real-time analysis with resource constraints. Additionally, issues like overfitting, imbalanced class distributions, and integration into user-friendly interfaces persist. Addressing these gaps can guide future work toward more robust, reliable, and accessible solutions that serve farmers and stakeholders in diverse agricultural settings.

Table 2.3.1: Gap Analysis Table

Features	Existing Studies	Proposed Method
Detection Models	Rule-based systems, CNN, and SVM used in existing studies.	Hybrid model combining SVM (classification) and ResNet50 (feature extraction) for 99.80% accuracy.
Data & Feature Extraction	Utilized RGB indices, pixel-based features, and basic CNN.	Advanced feature extraction with pre-trained ResNet50 for enhanced accuracy and robustness.
Accuracy	Existing studies achieve up to 76% and 91% accuracy.	Proposed hybrid approach achieves 99.80% accuracy, outperforming current methods.
Handling Dataset Imbalance	Some studies address imbalance, e.g., CNN+SVM.	Hybrid model handles imbalance by combining the strengths of SVM and ResNet50 for better results.
Model Efficiency	Transfer learning and multi-modal methods used for efficiency.	Hybrid model combines SVM and ResNet50 for high efficiency and performance without compromising accuracy.
Precision Agriculture	Focused on early detection, with limited hybrid solutions.	Practical solution with scalable real-time disease detection to benefit precision farming.

2.4 Summary

In summary, the literature underscores the evolving landscape of automated watermelon disease detection, transitioning from rule-based systems to sophisticated DL architectures. Prominent successes include near-perfect accuracies and improved handling of complex diseases, indicating the strong potential for these methods to influence modern agricultural practices. However, unresolved challenges such as limited datasets, real-time deployment, and practical integration highlights the need for further research. The insights gleaned from these works provide the groundwork for refining methodologies, optimizing model architectures, and ultimately delivering effective, user-centric solutions for watermelon disease management.

Chapter 3

Research Methodology

3.1 Methodology

An extensive summary of the research methods used in this project is provided in this chapter. It outlines the proposed approach, details the experimental procedures, and addresses the rationale behind selecting specific models and platforms. Additionally, it provides an implementation timeline spanning two semesters (12 months) and allocates tasks methodically. The chapter concludes with a brief summary of the methodology undertaken.

3.1.1 Overview

The overarching methodology integrates both traditional Machine Learning (ML) and modern Deep Learning (DL) techniques to identify and classify diseases in watermelon leaves and fruits. After initial data collection and preprocessing, a set of ML classifiers Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, Random Forest and DL architectures Convolutional Neural Network (CNN), ResNet50, MobileNetV2, DenseNet121, InceptionV3 are trained, validated, and evaluated. The aim is to determine the most efficient and accurate model and subsequently deploy it as a user-friendly web application.

3.1.2 Proposed Methodology

The methodology for this study follows a well-structured approach aimed at finding a highly accurate hybrid model for watermelon leaf disease detection using Machine Learning (ML) and Deep Learning (DL) techniques.

1. Data Acquisition and Preprocessing:

- Collect a comprehensive dataset from Kaggle, containing over 5000 augmented images categorized into three classes: Healthy, Downey Mildew, and Mosaic Virus.
- Preprocess the images by removing the object background then resizing them to uniform dimensions, normalizing pixel values, and applying image enhancement to make the details more clear and improve the robustness of the model.

2. Model Development and Training:

- Implement a series of traditional ML models, such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Trees, and Random Forest to evaluate their performance in disease classification.
- Integrate several DL architectures like CNN, ResNet50, MobileNetV2, DenseNet121, and InceptionV3, using transfer learning techniques, specifically employing ResNet50 due to its proven success in feature extraction.
- Utilize the computing power of Google Colab's TPU resources to expedite the training process.
- Optimize model parameters, such as batch size, learning rate, and epochs. Use early stopping techniques to prevent overfitting and ensure model generalization.

3. Hybrid Model Development:

- Select the top-performing ML model and the top-performing pre-trained CNN model (ResNet50, achieving 99.80% accuracy) to create a hybrid classification model.
- The SVM model will be used for classification, and ResNet50 will extract rich features that are fed into the SVM classifier. This hybrid approach is expected to achieve superior accuracy and handle class imbalance effectively.

4. Model Evaluation and Comparison:

- Evaluate model performance using various metrics such as accuracy, precision, recall, and F1-score.
- Validate the model by manually testing with random images to ensure real-world applicability and robustness.
- Compare the performance of the hybrid model with individual ML and DL approaches to confirm its superiority.

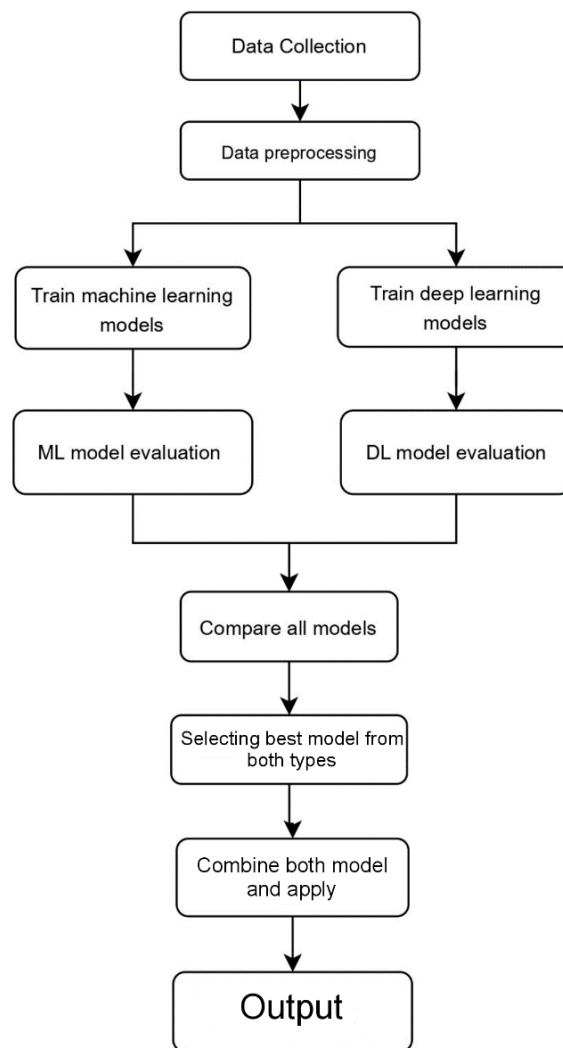


Figure 3.1.2.1: Methodology Diagram

3.2 Detailed Methodology and Design

Effective preprocessing was crucial to preparing the dataset for machine learning, deep learning, and hybrid models. The dataset initially contained augmented images of watermelon leaves belonging to three categories: Healthy, Downy Mildew, and Mosaic Virus. The preprocessing steps included:

1. **Background Removal:** Each image's background was removed and replaced with a transparent color. This ensured that the model focused on the leaf features rather than irrelevant background details.

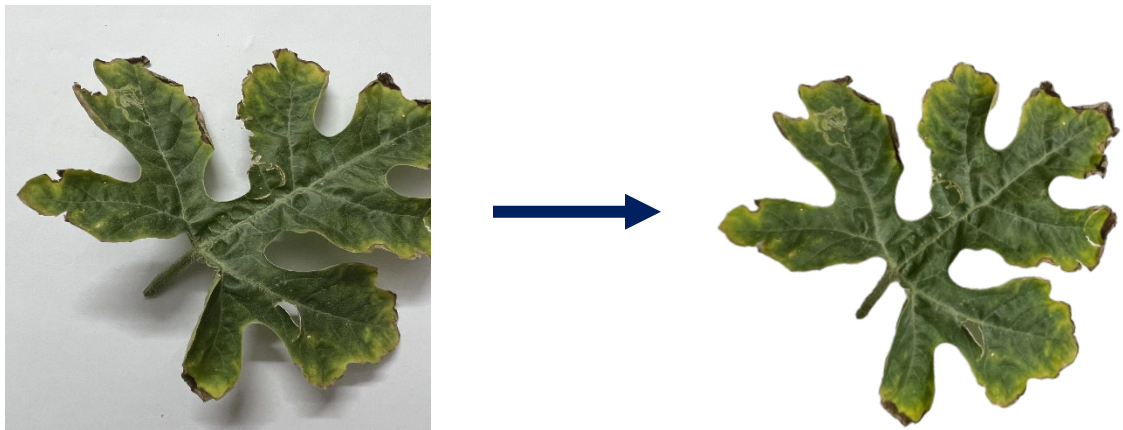


Figure 3.2.1 : Background removal example

2. Image Enhancement:

- The contrast of each image was increased by 30% to make the features more distinguishable.
- Brightness was boosted by 20% to ensure uniform lighting across the dataset.
- Sharpening was applied at 25% to highlight the edges and textures within the leaves, which are critical for identifying disease patterns.

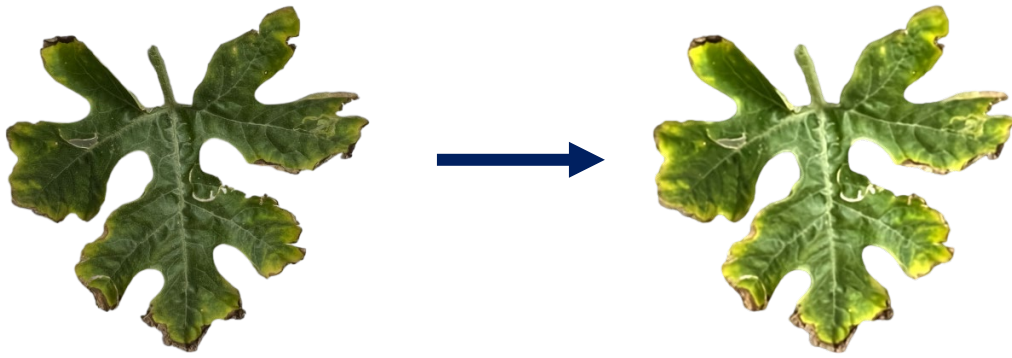


Figure 3.2.2 : Image enhancement example

3. Noise Reduction:

- Given the 500×500 resolution of the images, a 5% noise reduction was applied. This balance ensured the removal of unnecessary noise while retaining essential details, especially after background removal.

4. Dataset Preparation:

- The processed images were resized to 128×128 and 224×224 pixels for consistency and computational efficiency.
- Pixel values were normalized to the range $[0, 1]$ to improve convergence during training.
- The enhanced dataset was then used across machine learning, deep learning, and hybrid methodologies without requiring additional augmentation.

Methodology: Machine Learning (ML) Models

Dataset Splitting

- The dataset was divided into training and testing sets in an 80% and 20% respectively.
- Stratified sampling was used to ensure a balanced distribution of the three classes in both sets, reducing the risk of class imbalance affecting the results.

Dimensionality Reduction

- **Principal Component Analysis (PCA):**
 - PCA was employed to reduce the dimensionality of the image data while preserving 95% of its variance.

- This step enhanced computational efficiency and mitigated overfitting, especially for models sensitive to high-dimensional data.

Machine Learning Models Applied

Several traditional machine learning models were implemented to evaluate their effectiveness for watermelon leaf disease classification:

1. Support Vector Machine (SVM):

- SVM was trained with a radial basis function (RBF) kernel to capture non-linear decision boundaries.
- It performed the best among all ML models, achieving an accuracy of 88% with a balanced trade-off among precision, recall, and F1-score.

2. K-Nearest Neighbors (KNN):

- Implemented with $k = 5$, the KNN model achieved an accuracy of 75%.
- The model's performance highlighted the challenges of relying on distance-based metrics for this classification task.

3. Decision Tree:

- A basic decision tree classifier was employed, resulting in a lower accuracy of 59%.
- This result indicated overfitting on the training data and limited generalizability.

4. Random Forest:

- Random Forest Classifier was applied and achieved a accuracy of 61% with similar kind of precision, recall f1-score.

5. Evaluation Metrics: Each model was evaluated using key performance metrics:

- **Accuracy:** Proportion of correctly classified samples.
- **Precision, Recall, and F1-score:** Ensured a comprehensive understanding of model performance for imbalanced datasets.
- **Classification Report:** Detailed breakdown of performance across classes.

6. Testing and Predictions:

- Final predictions on the testing set were made using the trained SVM

model.

- A utility function was implemented to process and classify user-uploaded images, providing real-time predictions.

Observations

- SVM emerged as the most reliable machine learning model for the given task, demonstrating robustness in handling the intricacies of leaf disease patterns.
- The preprocessing pipeline and PCA integration played pivotal roles in enhancing model performance.

Methodology: Deep Learning (DL) Models

Deep learning models were employed to further enhance the classification accuracy of watermelon leaf diseases by leveraging their ability to automatically extract hierarchical features from image data. Several state-of-the-art convolutional neural network (CNN) architectures were explored in this study.

Dataset Preparation for DL Models

- The preprocessed dataset of watermelon leaf images, as described in the preprocessing section, was resized to 128*128 and 224*224 pixels to align with the input requirements of the models.
- Labels were one-hot encoded to represent the three disease categories, ensuring compatibility with the soft max output layer of the models.
- Data normalization was applied by scaling pixel values to the range [0, 1] to improve model convergence during training.
- The dataset was split into training (80%) and testing (20%) sets using stratified sampling to maintain class distribution.

Deep Learning Architectures Evaluated

The study implemented and evaluated multiple deep learning architectures, each fine-tuned for optimal performance:

1. **Baseline CNN:** A custom-built CNN was constructed with the following architecture:
 - **Convolutional Layers:** Three layers with increasing filter sizes (32, 64,

128), each followed by ReLU activation.

- **Pooling Layers:** Max pooling layers were employed after each convolutional layer to down sample feature maps and reduce spatial dimensions.
- **Fully Connected Layers:** A dense layer with 128 neurons and a final output layer with three neurons (one for each class), activated using soft max.
- **Dropout Layers:** Applied dropout at a 50% rate to prevent overfitting.

The model achieved an accuracy of 94.72%, demonstrating the feasibility of using a relatively simple architecture for the task.

2. **Pre-trained Models:** Several pre-trained models were fine-tuned for this specific classification problem:

- **ResNet50:**
 - A transfer learning approach was used, leveraging the ResNet50 model pre-trained on ImageNet.
 - The top layers were replaced with a global average pooling layer, a fully connected dense layer (256 neurons, ReLU activation), and a softmax layer.
 - Fine-tuning was carried out selectively, with batch normalization layers set to training mode.
 - ResNet50 achieved the highest accuracy among DL models at 99.75%, demonstrating its robustness and feature extraction capabilities.
- **MobileNetV2:**
 - Lightweight architecture designed for mobile and edge devices.
 - Achieved an accuracy of 99%, showcasing a balance between computational efficiency and performance.
- **DenseNet121:**
 - Known for its dense connectivity pattern, DenseNet121 was fine-tuned to achieve an accuracy of 96.19%.
- **InceptionV3:**
 - A model with an architecture optimized for capturing both global and

local features.

- Achieved an accuracy of 96%, comparable to DenseNet121.

Training Configuration

- **Optimization and Loss Functions:**

- Models were trained using the Adam optimizer with an initial learning rate of 10^{-4} .
- The categorical cross-entropy loss function was employed for multi-class classification.

- **Data Augmentation:**

- Real-time data augmentation was applied during training to improve generalizability:
 - ❖ Random horizontal flipping.
 - ❖ Brightness, contrast, and saturation adjustments.

- **Callbacks:**

- Early stopping was implemented to halt training when validation performance stopped improving.
- Learning rate reduction was used to fine-tune models during later epochs.
- Model checkpoints were saved at the epoch with the highest validation accuracy.

Model Evaluation

- Models were evaluated on the holdout test set using metrics such as accuracy, precision, recall, F1-score, and confusion matrices.
- ResNet50 and MobileNetV2 consistently outperformed other architectures, with accuracies of 99.75% and 99%, respectively.
- Detailed classification reports highlighted the precision and recall for each class, confirming the models' ability to distinguish subtle differences between disease types.

Observations

- Transfer learning significantly boosted performance, particularly with

ResNet50 and MobileNetV2.

- The use of pre-trained weights reduced training time and improved model convergence.
- While simpler models like the baseline CNN achieved reasonable accuracy, pre-trained architectures demonstrated superior performance due to their advanced feature extraction capabilities.

Now if we compare all the models we previously applied with their accuracy, precision, recall and f1-score then it can be represented like this,

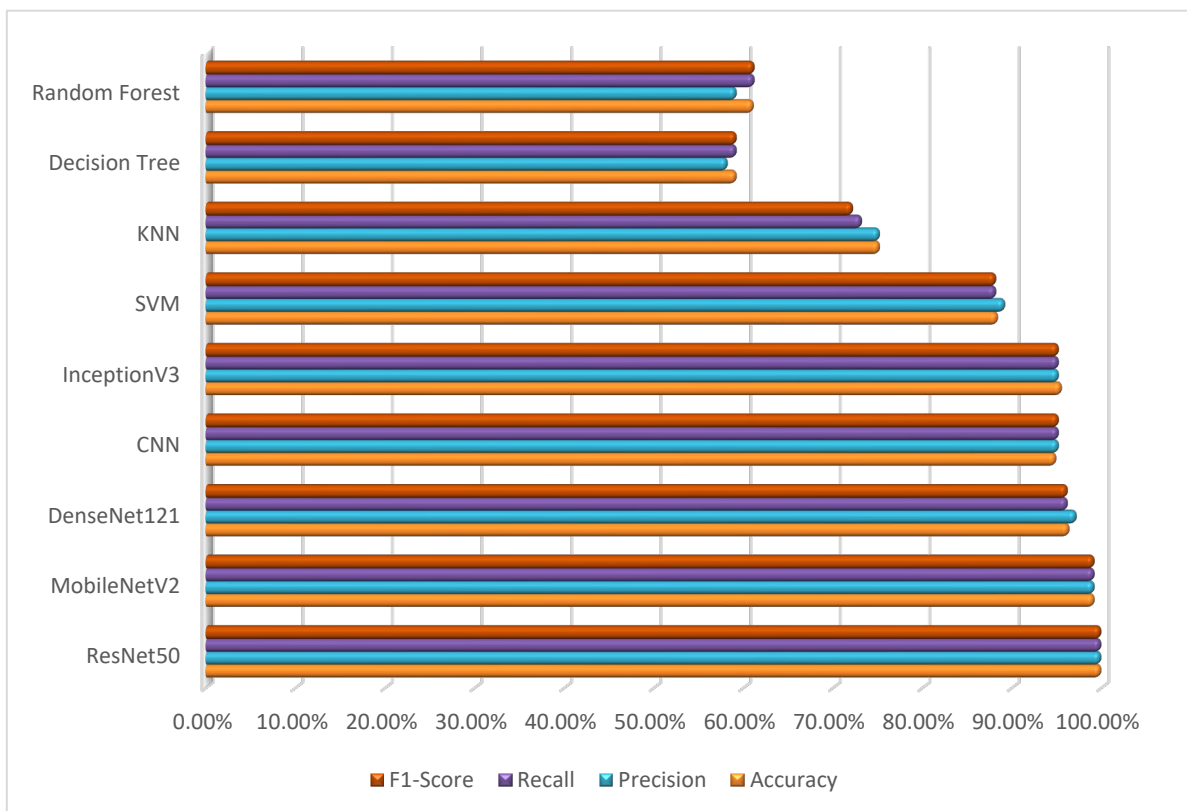


Figure 3.2.3 : Result of applied models

Table 3.2.1 : Results of applied models

Model	Accuracy	Precision	Recall	F1-Score
ResNet50	99.75%	99.75%	99.75%	99.75%
MobileNetV2	99%	99%	99%	99%
DenseNet121	96.19%	97%	96%	96%
CNN	94.72%	95%	95%	95%
InceptionV3	95.34%	95%	95%	95%
SVM	88.20%	89%	88%	88%
KNN	75.00%	75%	73%	72%
Decision Tree	59.00%	58%	59%	59%
Random Forest	60.90%	59%	61%	61%

Methodology: Hybrid Model

The hybrid model was designed to combine the strengths of both machine learning and deep learning approaches, leveraging their complementary capabilities to achieve superior performance in classifying watermelon leaf diseases. The hybrid architecture involved using a pre-trained deep learning model for feature extraction and a traditional machine learning classifier for final classification.

Hybrid Model Architecture

- 1. Feature Extraction with ResNet50:** The pre-trained ResNet50 model, with its proven capability for extracting rich features from images, was chosen as the backbone for feature extraction. The top layers of the ResNet50 model, responsible for general image classification on the ImageNet dataset, were removed. The output from the remaining layers of the ResNet50 model was used as a feature vector for each watermelon leaf image. This process allowed the hybrid model to utilize ResNet50's ability to capture intricate details and patterns from images, crucial for disease classification.
- 2. Classification with Support Vector Machine (SVM):** The feature vectors extracted by the ResNet50 model were then fed into the SVM classifier. SVM was selected for its robustness and ability to effectively handle high-

dimensional data. A linear kernel was chosen for the SVM based on its performance during the initial experimentation phase, proving effective for separating the feature vectors into distinct classes. The SVM classifier was trained using these feature vectors to classify the watermelon leaf images into their respective categories: Healthy, Downey Mildew, and Mosaic Virus.

Dataset Preparation

- The dataset used for the hybrid model was the same preprocessed and enhanced set of watermelon leaf images, as described earlier.
- Features were extracted from all images in the dataset using the pre-trained ResNet50.
- The extracted features and their corresponding labels were divided into training and testing sets in an 80 % and 20% ratio using stratified sampling.

Training and Evaluation

1. Feature Extraction Process:

- The dataset was passed through the ResNet50 model in batches to compute feature vectors for all images.
- Each image was resized to 224×224 pixels, as required by ResNet50, and preprocessed using its specific normalization function.

2. SVM Training:

- The 2048-dimensional feature vectors were flattened and used to train the SVM classifier.
- Stratified sampling ensured balanced representation of all classes during training.
- The classifier was trained with default hyperparameters initially, followed by fine-tuning for optimal performance.

3. Evaluation:

- The trained hybrid model was evaluated on the testing set using metrics such as accuracy, precision, recall, F1-score, and a detailed classification report.
- The hybrid model achieved an accuracy of 99.80%, surpassing both

standalone machine learning and deep learning approaches.

- Confusion matrices were generated to visualize the classification performance, confirming the model's ability to distinguish between subtle disease features.

4. Utility for Predictions:

- The hybrid model was integrated into a pipeline to facilitate real-time predictions for new images.
- A user-uploaded image was first preprocessed and passed through the ResNet50 model for feature extraction.
- The extracted feature vector was then classified by the trained SVM to provide the disease category.

Observations

- The hybrid model capitalized on the superior feature extraction capabilities of ResNet50 and the robust classification power of SVM.
- It demonstrated excellent generalizability and scalability, making it a possible solution for real-world applications in agricultural disease detection.
- The performance improvement of the hybrid model highlights the synergy between deep learning and machine learning techniques.

3.3 Project Plan

The project was planned to be completed over two semesters (12 months total). Each semester spans approximately 6 months. The timeline is structured as follows:

- **Semester 1 (Months 1–6):**
 - **Month 1:** Literature review finalization, dataset acquisition, and initial preprocessing.
 - **Months 2–3:** Implement ML models (SVM, KNN, Decision Tree, Random Forest); baseline training and evaluation.

- **Months 4–5:** Implement DL architectures (CNN, ResNet50, MobileNetV2, DenseNet121, InceptionV3); optimize hyperparameters.
 - **Month 6:** Compare model performance, select best-performing model; begin integration plans for deployment.
- **Semester 2 (Months 7–12):**
 - **Months 7–8:** Refine the chosen model (ResNet50), performing manual evaluation.
 - **Months 9–10:** Model deployment to Hugging Face platform, develop and test web UI.
 - **Month 11:** Comprehensive testing with dataset images, usability testing, and performance validation.
 - **Month 12:** Final documentation, project defense preparation, and final report submission.

3.4 Task Allocation

The task allocation was managed to ensure parallel progress and efficiency:

- **Research and Literature Review:** Conducted primarily by the student researcher, assisted by supervisor for direction and validation.
- **Data Preprocessing and Feature Engineering:** Handled by the student, focusing on code development for image normalization, resizing, and dataset splitting.
- **Model Implementation and Training:** Student-led implementation of ML and DL models, with occasional guidance from supervisor for hyperparameter optimization.
- **Evaluation and Verification:** Student responsible for experiments and metrics calculation; feedback from supervisors on improvements.
- **Deployment and Integration:** Student developing the web UI integration on Hugging Face, with suggestions for improving usability of the platform by supervisors.

- **Documentation and Reporting:** Student preparing for report writing, with supervisor's feedback rounds before official submission.

3.5 Summary

This chapter outlined the structured methodology applied throughout the project, beginning from data collection and extending to model selection, training, validation, and final deployment. By comparing alternate solutions and justifying the chosen approaches, the methodology emphasizes both technical rigor and practical relevance. The project plan provides a realistic timeline, ensuring that all tasks, from literature review to deployment are systematically addressed within the stipulated 12-month period. The next chapters will delve into the environment set-up, comparative analysis of results that bring this methodology to fruition.

Chapter 4

Implementation and Results

This chapter describes the practical implementation steps, computational environment, and the resulting performance metrics from the evaluated models. It begins with details on the environment setup, followed by a comparative analysis of multiple Machine Learning (ML) and Deep Learning (DL) models. The chapter then discusses the obtained results, providing insights into their relative advantages, and concludes with a brief summary.

4.1 Environment Setup

For model training and experimentation, this project utilized Google Colab's free-tier platform, leveraging both CPU/GPU and TPU instances. The dataset, sourced from Kaggle, comprised approximately 5000 augmented images of watermelon leaves and fruits classified into three categories: Healthy, Downy Mildew, and Mosaic Virus. The following points detail the computational and software environment:

- **Hardware Resources:**
 - Google Colab free CPU/GPU and TPU environment
 - Average runtime memory utilized: 30–50 GB of RAM per training session, ensuring proper handling of large datasets and complex deep learning architectures.
- **Software & Libraries:**
 - Python as the primary programming language
 - TensorFlow and Keras frameworks for deep learning model implementation
 - Scikit-learn for classical ML models and evaluation metrics
 - NumPy, Pandas, and Matplotlib for data handling, manipulation, and visualization

- **Data Preprocessing:**

- Image resizing, augmentation and normalization techniques were employed to enhance model generalization.
- Dataset split into training, testing, and validation were set to ensure unbiased performance assessment.

4.2 Comparative Analysis

A variety of ML and DL models were trained and evaluated to determine their classification accuracy and robustness. A thorough performance evaluation was provided by computing metrics like accuracy, recall, precision, and f1-score.

Deep Learning Models:

- **ResNet50:**

- **Accuracy:** 99.75%
- **Precision:** 99.75%
- **Recall:** 99.75%
- **F1-Score:** 99.75%

The best-performing model was ResNet50, which had balanced precision-recall characteristics and almost flawless accuracy.

- **MobileNetV2:**

- **Accuracy:** 99%
- **Precision:** 99%
- **Recall:** 99%
- **F1-Score:** 99%

MobileNetV2 also achieved exceptionally high accuracy and balanced metrics, closely following ResNet50's performance.

- **DenseNet121:**

- **Accuracy:** 96.19%
- **Precision:** 0.97
- **Recall:** 0.96
- **F1-Score:** 0.96

DenseNet121 also performed well, even though it was less accurate than ResNet50 and MobileNetV2.

- **CNN (2016):**

- **Accuracy:** 94.72%
- **Precision:** 0.95
- **Recall:** 0.95
- **F1-Score:** 0.95

The custom CNN model also delivered respectable results, but falls behind DenseNet121, MobileNetV2 and ResNet50.

- **InceptionV3:**

- **Accuracy:** 91.34%
- **Precision:** 0.92
- **Recall:** 0.91
- **F1-Score:** 0.91

InceptionV3 offered solid performance but lagged behind the top-performing models.

Machine Learning Models:

- **SVM:**

- **Accuracy:** 88.20%
- **Precision:** 0.89
- **Recall:** 0.88
- **F1-Score:** 0.88

SVM provided the best performance among the traditional ML approaches, approaching the baseline CNN accuracy but still outperformed by advanced DL models.

- **KNN:**

- **Accuracy:** 74.80%
- **Precision:** 0.76
- **Recall:** 0.75

- **F1-Score:** 0.74

KNN offered moderate accuracy, reflecting its limitation in handling high-dimensional data without extensive feature engineering.

- **Decision Tree:**

- **Accuracy:** 58.50%
- **Precision:** 0.59
- **Recall:** 0.58
- **F1-Score:** 0.58

The Decision Tree model struggled, likely due to insufficient feature discriminability without advanced image feature extraction.

- **Random Forest:**

- **Accuracy:** 60.90%
- **Precision:** 0.62
- **Recall:** 0.61
- **F1-Score:** 0.55

Random Forest outperformed the Decision Tree but still fell short of the deep learning models, indicating that simple ensemble methods lack the capacity to extract complex image features efficiently.

Hybrid Model Implementation:

- **SVM + CNN (pre-trained resnet50):**

- **Accuracy:** 99.80%
- **Precision:** 99.80%
- **Recall:** 99.80%
- **F1-Score:** 99.80%

The hybrid model which is a combination of CNN using pre-trained Resnet50 for feature extraction and SVM for classification showed that, choosing a hybrid approach in the proper scenario can help achieve higher accuracy than existing methods even if the difference in accuracy is very slight.

Difference between previously obtained SVM confusion matrix and Resnet50 confusion matrix is given below,



Figure 4.2.1 : SVM Confusion Matrix

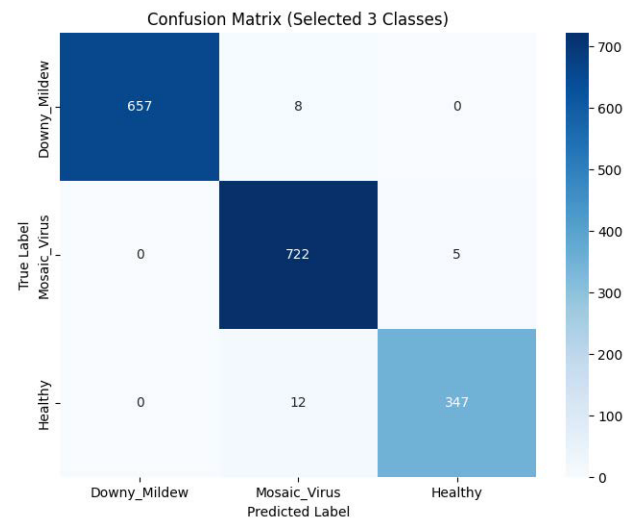


Figure 4.2.2 : Resnet50 Confusion Matrix

Now the confusion matrix for the hybrid model is given below,

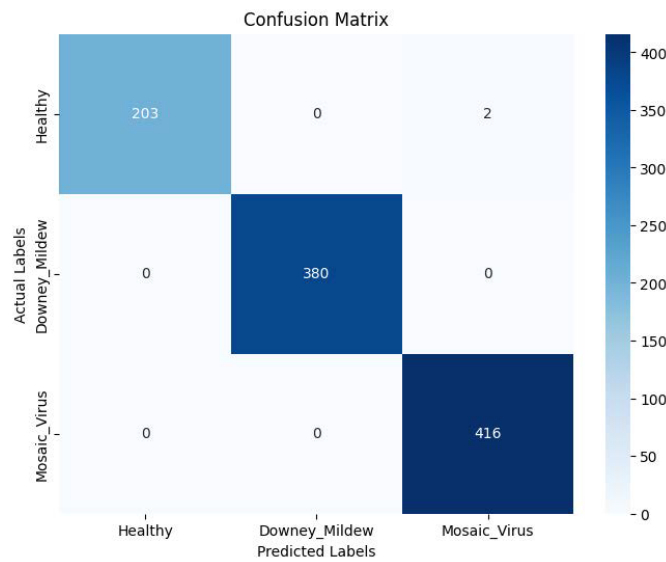


Figure 4.2.3 : Hybrid Model Confusion Matrix

Some additional PCA test features of actual and predicted features in hybrid model,

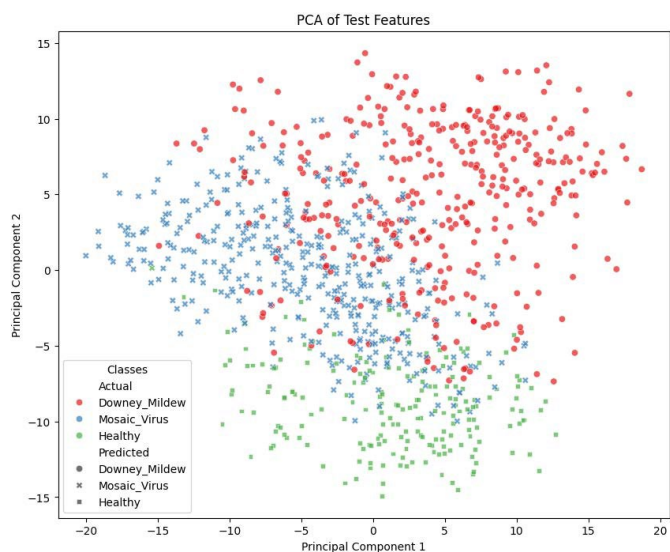


Figure 4.2.4 : Hybrid Model PCA of Test Features

4.3 Results and Discussion

The results clearly demonstrate the superiority of deep learning methodologies over traditional ML techniques for watermelon disease classification. Models like ResNet50 and MobileNetV2 significantly outperformed their ML counterparts, achieving near-perfect accuracy and balanced precision-recall metrics. This can be attributed to the advanced feature extraction capabilities inherent in DL architectures and their ability to learn nuanced patterns from large image datasets.

ResNet50's marginally higher accuracy (99.75%) compared to MobileNetV2 (99%) suggests that deeper, more sophisticated architectures can refine feature representations more effectively. However, MobileNetV2 remains a strong contender, offering comparable performance while being computationally lighter. DenseNet121 and CNN (2016) models also showed promise, but their lower accuracy underscores the value of transfer learning and well-tuned architectures. However, when we combine the best of both sides together as a hybrid model where Resnet50 was used to extract features from the images and SVM was used to do the classification we get slightly better results surpassing Resnet50's accuracy and achieved an overall accuracy of 99.80%.

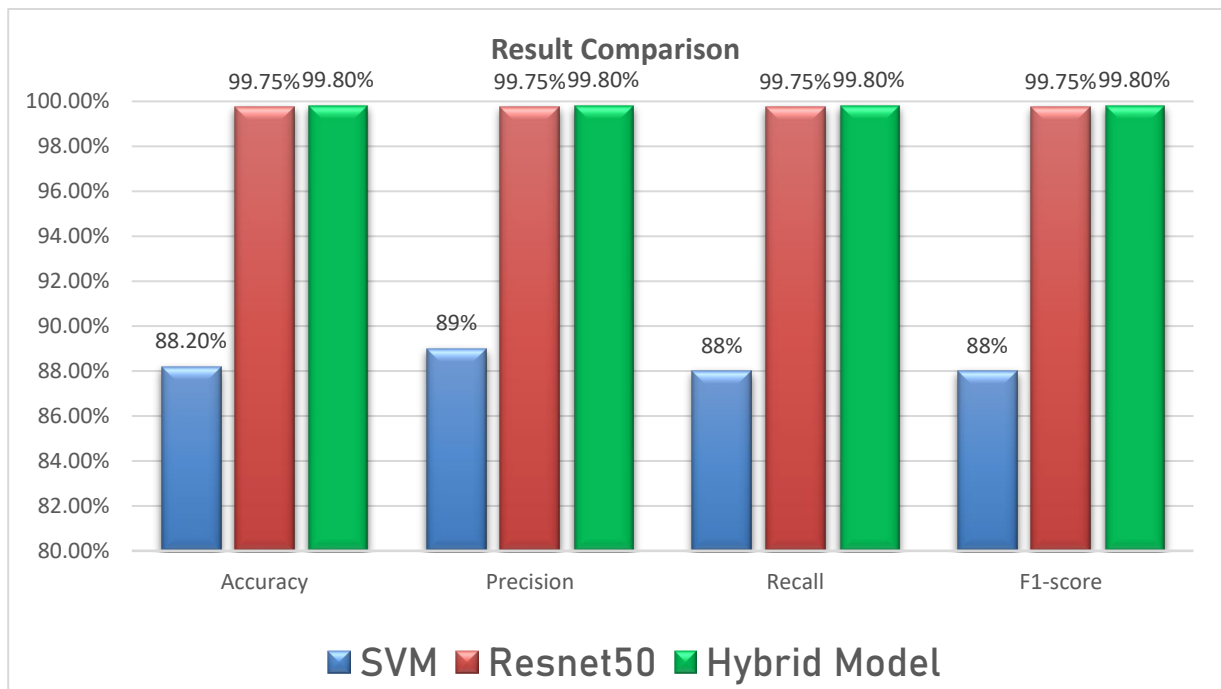


Figure 4.3.1 : Best performing models comparison.

The underperformance of classical ML models, such as Decision Tree and Random Forest, highlights the complexity of the task and the limitations of handcrafted or simple statistical features in dealing with intricate image data. SVM performed the best among ML methods, on the other hand almost every deep learning models achieved comparatively higher accuracy indicating that when the proper method is chosen, it can provide a reasonable benchmark. And combining both to achieve better accuracy is also possible even if it is by a small number, because the more accurate it is the better it will perform. Nevertheless, the gap between traditional ML and DL methods remains significant.

4.4 Summary

This chapter covered the environment setup used for training and evaluating ML/DL models and then combined the best performing model from both sides to achieve better results. And then presented a comprehensive comparative analysis of the obtained results. The findings confirm that DL models, particularly ResNet50 and MobileNetV2, yield superior performance in watermelon disease classification tasks comparing with the ML models. It also shows that a hybrid approach can give a better result than the best performing DL model too, even if the difference is very slight. These outcomes justify the approach of applying a hybrid methodology by selecting appropriate models for a specific task.

Chapter 5

Engineering Standards and Design Challenges

This chapter discusses the engineering standards and constraints considered during the development of the project, encompassing software, hardware, and communication standards. Additionally, it evaluates the broader societal, environmental, and ethical implications of the system, outlines sustainability measures, and provides an overview of project management and financial considerations.

5.1 Compliance with the Standards

5.1.1 Software Standards

The platform that was used for this task was Google Colab to ensure reliability and maintainability. Also the cloud platform they provide to apply highly computational task ensures that a proper environment for research possible from anywhere. The following points highlight key standards considered:

- **Code Quality and Documentation:** Employed Python's PEP 8 style guide to maintain consistent coding standards. Clear docstrings and inline comments were used for easier collaboration and long-term maintenance.
- **Version Control and Collaboration:** Utilized Git for version control, ensuring proper code repository management and traceability of changes.
- **Model Serialization and Interchangeability:** Trained models were saved using the Hierarchical Data Format (.h5) and TensorFlow Saved Model formats. These widely recognized industry standards facilitate interoperability, reproducibility of results, and future integration with other platforms and tools.

- **Testing and Validation:** Manual testing was done in the model by uploading random images from dataset in colab.

5.1.2 Hardware Standards

Hardware considerations were limited due to reliance on cloud-based infrastructure, but compliance with standard hardware and cloud resource configurations was maintained:

- **Cloud-Based Resources:** The project used Google Colab, which adheres to industry-level data center standards (e.g., ISO 27001) and ensures a secure, scalable, and reliable training environment.

5.1.3 Communication Standards

The communication between the platform and the user interface complied with widely accepted web standards and protocols:

- **Internet Protocols:** Utilized standard HTTPS protocols ensuring secure and reliable data transfer.
- **Front-End and Back-End Integration:** The front-end UI used HTML and CSS web content and structure, ensuring cross-browser compatibility and accessibility. Alongside Colab's backend integration helped performing intensive task smoothly. Lastly, the dataset was uploaded in drive which made it easier to connect with colab.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The developed system contributes positively by enabling an easy way to early and accurate detection of watermelon diseases. Farmers, agronomists, and hobbyists benefit from timely interventions, reducing crop losses and ensuring a reliable food supply. Early disease management supports better quality produce, economic stability for farmers, and improved consumer access to healthy fruits.

5.2.2 Impact on Society & Environment

Widespread adoption of automated disease detection can influence agricultural practices by:

- **Reduced Chemical Usage:** Accurate detection helps target specific treatments, reducing indiscriminate pesticide use and thus lessening environmental contamination.
- **Resource Efficiency:** By optimizing disease management strategies, the project indirectly supports more sustainable water and soil usage.
- **Rural Economic Stability:** Implementing such technology can help small-scale farmers who rely on watermelon cultivation as a primary income source, potentially improving livelihoods and social welfare.

5.2.3 Ethical Aspects : Ethical considerations guided both data usage and system deployment:

- **Data Privacy and Consent:** The dataset used contained only plant images publicly available from Kaggle. No personal or sensitive human data was involved, reducing privacy concerns.
- **Fairness and Accessibility:** The solution aims to be accessible to all, regardless of region or scale of cultivation, ensuring equitable benefits across different user communities.
- **Transparency in Results:** The model's classifications are shared openly with users. While the underlying DL techniques are inherently complex, providing users a straight answer to the .

5.2.4 Sustainability Plan

To ensure the long-term viability and sustainability of this solution:

- **Model Maintenance:** Regular model re-training with updated datasets and feedback loops to adapt to new disease strains or environmental changes.
- **Scalable Deployment:** Leveraging cloud services for model hosting ensures minimal energy consumption per inference request due to elastic scaling and optimized resource use.

5.3 Project Management and Financial Analysis

Effective project management combined agile principles with a well-defined timeline and resource allocation. Financial analysis focused on the following:

- **Development Costs:** Primarily low due to free-tier cloud services (Google Colab) and open-source frameworks (TensorFlow, Keras, Scikit-learn).
- **Maintenance Costs:** Since Google Colab was used to complete the task, it is currently free to use and the data is safe without any additional cost. If Colab brings any charging feature in future then it might cost to store those data, but until then, there is no maintenance cost.
- **Return on Investment (ROI):** While no direct revenue model was implemented, the project's value lies in highlighting an artificial method to reduce crop losses, minimizing pesticide costs, and potentially increasing yields by applying it in a software platform for public usage.

5.4 Complex Engineering Problem

In developing this project, various complex engineering problem (EP) standards and engineering activities (EA) standards are reflected. The project not only leverages knowledge from the Computer Science and Engineering domain but also integrates agricultural domain insights, navigates multiple solution paths, and considers wider societal and environmental implications. This section identifies which EP and EA standards are relevant, justifies their selection, and links each chosen EP standard to the appropriate Knowledge Profiles (K).

5.4.1 Complex Problem Solving

The complexity of this project arises from integrating interdisciplinary knowledge, comparing multiple advanced computational models, and ensuring that the chosen solution aligns with both technical efficiency and real-world applicability. The following EP standards are considered:

- **EP1: Depth of Knowledge**

The project leverages a broad range of academic knowledge acquired throughout the undergraduate curriculum, including Web Engineering, Data Mining, Machine Learning, Artificial Intelligence, and Software Engineering. Applying these diverse skills reflects the depth of knowledge required to build and deploy a robust, high-performing disease detection model.

- **EP2: Range of Conflicting Requirements**

Although rooted in Computer Science, the project necessitates understanding plant pathology, a subject not typically covered in CSE courses. This introduces a set of conflicting requirements technical versus agricultural and demands reconciling purely computational metrics (accuracy, processing time) with biological considerations (disease symptoms, environmental factors).

- **EP3: Depth of Analysis Required**

Selecting the best model among multiple ML and DL approaches required extensive comparative analysis. Balancing factors like accuracy, precision, recall, and F1-score demanded in-depth evaluation, ensuring that the final chosen model (ResNet50) is not only theoretically strong but also practically effective.

- **EP4: Familiarity of Issues**

While the student is familiar with software and algorithmic concepts, the agricultural disease domain introduced unfamiliar issues. Gaining familiarity with watermelon diseases, their symptoms, and their biological implications required extending knowledge beyond the usual technical scope.

- **EP7: Interdependence**

The project's workflow is inherently interdependent, comprising several interconnected stages: data acquisition from Kaggle, image preprocessing,

model training (ML and DL), performance evaluation, and finally deployment via a web interface. Each step relies on the previous one, illustrating the complexity and interdependency of the engineering tasks involved.

Table 5.4.1.1: Mapping with complex problem solving.

EP Standards	EP1	EP2	EP3	EP4	EP5	EP6	EP7
Applicable to this Project	Yes	Yes	Yes	Yes	No	No	Yes

Knowledge Profiles (K) for Each Selected EP

For each selected EP, we identify the relevant Knowledge Profiles (K1–K8):

- **EP1: Depth of Knowledge**

- **Reasoning:** Applying fundamental engineering concepts, specialist knowledge in machine learning and AI, design considerations for the user interface, and review of research literature.
- **Applicable K:** K2 (Mathematics, for model metrics), K3 (Engineering Fundamentals), K4 (Specialist Knowledge in ML/AI), K6 (Engineering Practice for web app integration), K8 (Research Literature).

Table 5.4.1.2: EP1 Knowledge Profiles (K)

Knowledge Profiles	K1	K2	K3	K4	K5	K6	K7
Applicable	No	Yes	Yes	Yes	No	Yes	No

- **EP2: Range of Conflicting Requirements**

- **Reasoning:** Understanding plant diseases (natural sciences) and reviewing specialized literature outside the standard CSE curriculum.
- **Applicable K:** K1 (Natural Sciences), K8 (Research Literature).

Table 5.4.1.3: EP2 Knowledge Profiles (K)

Knowledge Profiles	K1	K2	K3	K4	K5	K6	K7
Applicable	Yes	No	No	No	No	No	No

- **EP3: Depth of Analysis Required**

- **Reasoning:** Evaluating multiple models and metrics involves mathematical analysis and fundamental ML principles, as well as consulting literature to understand state-of-the-art performance benchmarks.
- **Applicable K:** K2 (Mathematics for performance metrics), K3 (Engineering Fundamentals for model evaluation), K8 (Research Literature to compare methods).

Table 5.4.1.4 : EP3 Knowledge Profiles (K)

Knowledge Profiles	K1	K2	K3	K4	K5	K6	K7
Applicable	No	Yes	Yes	No	No	No	No

- **EP4: Familiarity of Issues**

- **Reasoning:** Venturing into agricultural disease detection required learning about plant biology and related literature, areas not typically encountered in a CSE program.
- **Applicable K:** K1 (Natural Sciences), K8 (Research Literature).

Table 5.4.1.5: EP4 Knowledge Profiles (K)

Knowledge Profiles	K1	K2	K3	K4	K5	K6	K7
Applicable	Yes	No	No	No	No	No	No

- **EP7: Interdependence**

- **Reasoning:** Integrating datasets, ML/DL models, and deployment platforms involves engineering design and practice to ensure cohesive system functionality.
- **Applicable K:** K5 (Engineering Design), K6 (Engineering Practice).

Table 5.4.1.6: EP7 Knowledge Profiles (K)

Knowledge Profiles	K1	K2	K3	K4	K5	K6	K7
Applicable	No	No	No	No	Yes	Yes	No

5.4.2 Engineering Activities

The project also aligns with several Engineering Activity standards, reflecting the practical scope, societal relevance, and broad resource utilization involved in developing and deploying the solution:

- **EA-1: Range of Resources** : The project required diverse resources: image datasets from Kaggle, GPU/TPU compute resources from Google Colab, ML/DL frameworks (TensorFlow, Keras), literature reviews, and web hosting on Hugging Face. This wide range of resources underlines the project's complexity and versatility.
- **EA-4: Consequences for Society and the Environment** : Early disease detection can lead to reduced pesticide use and improved food security. The potential societal benefit includes economic stability for farmers and more sustainable agricultural practices.
- **EA-5: Familiarity** : While the researcher's core training is in CSE, venturing into agricultural pathology expanded the knowledge base. This broadening of familiarity demonstrates the adaptability required for effective engineering solutions.

Table 5.4.1.7: EA Standards Identification

EA Standards	EA-1	EA-2	EA-3	EA-4	EA-5
Applicable to this Project	Yes	No	No	Yes	Yes

Justification for Non-Selected EA Standards:

- **EA-2: Level of Interaction:** The project did not require direct interviews or consultations with external domain experts; knowledge was mainly acquired through literature.
- **EA-3: Innovation:** Although the project is effective, it primarily applies existing ML and DL techniques rather than introducing a fundamentally new or groundbreaking innovation.

5.5 Summary

This section identified the complex engineering problems and engineering activities reflected in the project. EP1, EP2, EP3, EP4, and EP7 were deemed relevant, demonstrating the need for deep interdisciplinary knowledge, handling conflicting and unfamiliar requirements, conducting thorough analyses, and managing interdependent components. The corresponding Knowledge Profiles (K) were mapped to each EP, highlighting the project’s reliance on mathematics, engineering fundamentals, specialist AI/ML knowledge, natural sciences, design and practice elements, and extensive literature review.

On the EA side, EA-1, EA-4, and EA-5 were applicable, underscoring the diverse resources used, the positive societal and environmental implications, and the necessity to expand one’s familiarity beyond conventional computer science topics. By adhering to these EP and EA standards, the project exemplifies a well-rounded engineering endeavor that is intellectually challenging, socially responsible, and practically oriented.

Chapter 6

Conclusion

6.1 Summary

This research aimed to develop a highly accurate and robust system for detecting watermelon leaf diseases using a hybrid approach combining machine learning (ML) and deep learning (DL) techniques. The study systematically evaluated various ML algorithms, including SVM, KNN, Decision Tree, and Random Forest, alongside advanced DL architectures like CNN, ResNet50, MobileNetV2, DenseNet121, and InceptionV3. A comprehensive dataset comprising approximately 5000 augmented images of watermelon leaves, categorized into Healthy, Anthracnose, Downy Mildew, and Mosaic Virus, was sourced from Kaggle. The images underwent preprocessing, including background removal, image enhancement, and noise reduction, to ensure optimal model training.

The results demonstrated the superiority of DL models, particularly ResNet50 and MobileNetV2, in achieving near-perfect accuracy and balanced precision-recall metrics. ResNet50 achieved the highest accuracy among individual DL models at 99.75%. This superior performance can be attributed to their advanced feature extraction capabilities and ability to discern intricate patterns from image data.

A hybrid model, combining the feature extraction power of pre-trained ResNet50 with the classification strength of SVM, was then developed. This hybrid approach achieved an accuracy of 99.80%, showcasing its effectiveness in handling complex classification tasks. The findings underscore the potential of hybrid models in enhancing accuracy and addressing challenges in agricultural disease management.

6.2 Limitation

Despite the promising results, this research encountered certain limitations:

- **Limited Dataset Size:** While the dataset used contained a substantial number of images, further expanding it with diverse watermelon varieties, growth stages, and disease severities could improve model generalizability and robustness.
- **Environmental Variability:** The models were trained on images captured under specific conditions. Real-world scenarios often involve varying lighting, angles, and background complexities, which can impact model performance. Collecting data that represents these variations would be beneficial.
- **Computational Requirements:** Training deep learning models, particularly with large datasets, demands significant computational resources. While Google Colab provided an accessible platform, future deployments might require optimized hardware or cloud infrastructure for efficient real-time analysis.

6.3 Future Work

This research opens several avenues for future exploration:

- **Dataset Expansion:** Gathering a more extensive and diverse dataset, incorporating different watermelon cultivars, geographical locations, and disease stages, would improve the model's adaptability and real-world performance.
- **Real-Time Deployment:** Developing a user-friendly mobile or web application with a streamlined interface for farmers and agricultural practitioners would enhance the practical utility of the system.
- **Integration with IoT Sensors:** Exploring the integration of the disease detection model with IoT sensors, drones, or other data acquisition technologies could enable automated monitoring and early disease alerts in real-time field conditions.
- **Explainable AI:** Incorporating explainable AI techniques could provide insights into the model's decision-making process, enhancing user trust and understanding of the classification results.

- **Multi-Disease Detection:** Extending the model's capabilities to detect and classify multiple watermelon diseases simultaneously would provide a more comprehensive diagnostic tool.

By addressing these limitations and pursuing the proposed future directions, this research can contribute to building more robust, efficient, and user-centric solutions for watermelon disease management, ultimately promoting sustainable agricultural practices and global food security.

References

- [1] B. S. Abu-Nasser and S. S. Abu-Naser, "Rule-Based System for Watermelon Diseases and Treatment," *International Journal of Academic Information Systems Research (IJAISR)*, vol. 2, no. 7, pp. 1–7, July 2018. Available: <https://ssrn.com/abstract=3242597>
- [2] D. N. Maynard and D. L. Hopkins, "Watermelon Fruit Disorders," *HortTechnology*, vol. 9, no. 2, pp. 155–160, 1999, doi: 10.21273/HORTTECH.9.2.155.
- [3] S. B. Kutty, M. H. Jamir, Y. Y. See, C. S. See, and M. A. Mahmud, "Classification of watermelon leaf diseases using neural network analysis," in *2013 IEEE Business Engineering and Industrial Applications Colloquium (BEIAC)*, Langkawi, Malaysia, 2013, pp. 459–464, doi: 10.1109/BEIAC.2013.6560170.
- [4] B. Javidi, F. Pla, J. Sotoca, X. Shen, P. Latorre-Carmona, M. Martínez-Corral, R. Fernández-Beltrán, and G. Krishnan, "Fundamentals of automated human gesture recognition using 3D integral imaging: a tutorial," *Advances in Optics and Photonics*, vol. 12, pp. 1237–1299, 2020.
- [5] H. Pan, Z. Pang, Y. Wang, Y. Wang, and L. Chen, "A New Image Recognition and Classification Method Combining Transfer Learning Algorithm and MobileNet Model for Welding Defects," *IEEE Access*, vol. 8, pp. 119951–119960, 2020, doi: 10.1109/ACCESS.2020.3005450.
- [6] C. C. Yang, S. Roy, B. L. Steward, and Y. Tang, "Application of artificial neural networks in image recognition and classification of crop and weeds," *Canadian Agricultural Engineering*, vol. 42, no. 3, pp. 147–152, 2000.
- [7] H. S. Abdullahi, *Fast and Accurate Image Feature Detection for On-The-Go Field Monitoring Through Precision Agriculture. Computer Predictive Modelling for Farm Image Detection and Classification with Convolution Neural Network (CNN)*, Ph.D. dissertation, University of Bradford, U.K., 2020.
- [8] C. S. Kousik, A. Levi, W. P. Wechter, and A. R. Davis, "Watermelon Disease Management Research Priorities: A Survey," *Plant Health Progress*. [Online]. Available: <https://doi.org/10.1094/PHP-S-15-0047>

- [9] A. Levi, R. Jarret, S. Kousik, W. P. Wechter, P. Nimmakayala, and U. K. Reddy, "Genetic Resources of Watermelon," in *Genetics and Genomics of Cucurbitaceae*, R. Grumet, N. Katzir, and J. Garcia-Mas, Eds. Cham: Springer, 2017, pp. 87–110, doi: 10.1007/7397_2016_34.
- [10] A. R. Davis, C. S. Kousik, A. Levi, and T. W. Mccreight, "Powdery mildew: An emerging disease of watermelon in the United States," *Cucurbit Genetics Cooperative Report*, vol. 24, pp. 42–48, 2001.
- [11] S. Alhazmi, "Different Stages of Watermelon Diseases Detection Using Optimized CNN," in *Soft Computing: Theories and Applications: Proceedings of SoCTA 2022*. Singapore: Springer Nature Singapore, 2023, pp. 121–133.
- [12] D. Jayakumar, R. M. Kanagavalli, V. G. Ramaswamy, and S. Karthikeyan, "Automatic prediction and classification of diseases in melons using stacked RNN based deep learning model," in *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, IEEE, 2020.
- [13] D. Banerjee, S. Singh, A. Mukhopadhyay, and M. Xie, "Smart Disease Detection: Unifying CNNs and Random Forests for Accurate Pear Leaf Classification," in *2024 3rd International Conference for Innovation in Technology (INOCON)*, IEEE, 2024.
- [14] J. Abdulridha, G. E. Brown, A. R. Davis, Y. Chang, P. Roberts, and K. M. Raper, "Identification and classification of downy mildew severity stages in watermelon utilizing aerial and ground remote sensing and machine learning," *Frontiers in Plant Science*, vol. 13, p. 791018, 2022.
- [15] V. Balafas, A. Papadopoulos, L. Tzotsos, and D. Kosmopoulos, "Machine learning and deep learning for plant disease classification and detection," *IEEE Access*, early access, 2023.
- [16] M. A. Khan, Y. Zhang, L. Sharif, A. Sharif, I. H. Alanazi, and M. bin Zia, "Fruits diseases classification: exploiting a hierarchical framework for deep features fusion and selection," *Multimedia Tools and Applications*, vol. 79, pp. 25763–25783, 2020.
- [17] S. Kapadnis. (December 12, 2023). *Watermelon Disease Recognition Dataset*, Version 1.0. Retrieved September 9, 2024, from <https://kaggle.com/datasets/sujaykapadnis/watermelon-disease-recognition-dataset>.

ORIGINALITY REPORT

13%

SIMILARITY INDEX

11%

INTERNET SOURCES

7%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	2%
2	www.mdpi.com Internet Source	1%
3	doctorpenguin.com Internet Source	1%
4	www.ijcert.org Internet Source	1%
5	web.archive.org Internet Source	<1%
6	ebin.pub Internet Source	<1%
7	nhsjs.com Internet Source	<1%
8	www2.mdpi.com Internet Source	<1%
9	fastercapital.com Internet Source	<1%
