Smart Agriculture for Peanuts: Advanced Techniques for Classification, Disease Detection, and Pest Attack Identification

DEPARTMENT OF SOFTWARE ENGINEERING
UNIVERSITY OF SARGODHA
SARGODHA – PAKISTAN

SESSION 2021-2025

Smart Agriculture for Peanuts: Advanced Techniques for Classification, Disease Detection, and Pest Attack Identification

Undertaken By:

STUDENT NAME Kashif Mahmood

ROLL NO. BSSE51F21S007

STUDENT NAME: Nabeel Abbas

ROLL NO. BSSE51F21S026

STUDENT NAME: Ume Habiba

ROLL NO. BSSE51F21S029

Supervised By:

SUPERVISOR NAME: Dr. Muhammad Ramzan

SIGNATURE_______________

A DISSERTATION SUBMITTED AS A PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELORS IN COMPUTER SCIENCE / SOFTWARE ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING
UNIVERSITY OF SARGODHA
SARGODHA – PAKISTAN

SESSION 2021-2025

FINAL APPROVAL

The department will provide the final approval page after completing the final evaluation.

DEDICATION

I sincerely dedicate this project to my beloved parents, whose unwavering support, love, and encouragement have been the foundation of my journey. Their faith in my abilities has been my greatest source of motivation.

I also express my heartfelt gratitude to my teachers, who have shared invaluable knowledge and guidance throughout my academic path. Their mentorship and inspiration have been instrumental in developing my skills and understanding, making this project a reality.

This work is a tribute to my parents and teachers' profound influence on my life, and I humbly present it as a token of my appreciation for their endless sacrifices and guidance.

ACKNOWLEDGEMENT

I am deeply thankful to the Almighty for granting me the wisdom, strength and perseverance needed to undertake and complete this project.

I extend my heartfelt gratitude to my parents, whose unwavering support, encouragement, and guidance which have been the foundation of my academic and personal growth. Their sacrifices and faith in my abilities remain a constant source of inspiration.

I wish to express my sincere thanks to my teachers and mentors for their expertise, patience, and invaluable guidance throughout this journey. Their constructive feedback and intellectual insights have enriched this project and depended my understanding of the subject.

Additionally, I am grateful to my peers and colleagues for their encouragement and engaging discussions, which have played a key role in refining my ideas. Lastly, I extend my appreciation to my institution for providing the resources and platform that made this endeavor possible.

PROJECT BRIEF

PROJECT NAME	Smart Agriculture for Peanuts: Advanced Techniques for Classification, Disease Detection, and Pest Attack Identification
ORGANIZATION NAME	UNIVERSITY OF SARGODHA
OBJECTIVE	TO BUILD A SMART AGRICULTURE SYSTEM FOR THE DETECTION OF PEANUT DISEASES, PEANUT PESTS, AND PEANUT CLASSIFICATION.
UNDERTAKEN BY	UME HABIBA KASHIF MAHMOOD NABEEL ABBAS
SUPERVISED BY	DR. MUHAMMAD RAMZAN ASSISTANT PROFESSOR UNIVERSITY OF SARGODHA
STARTED ON	2/9/2024
COMPLETED ON	23/5/2025
OPERATING SYSTEM	WINDOWS
TOOLS USED	Html, CSS, JavaScript, Bootstrap, Flask (Python), PHP, MYSQL

ABSTRACT

Many economies depend on agriculture for food and because it greatly contributes to their income. Peanuts are very important from a financial standpoint among crops and oilseeds play a key role in the economy. Peanut Classification, Peanut Diseases and Peanut Pests are some of the areas we have covered. Our main goal is to spot the current technology weaknesses and plan the next steps for research that can help build better, more practical and efficient methods for managing peanut farming. Because different types of peanuts can affect yield and quality, it is necessary to correctly identify them. We pay special attention to analyzing available computational approaches, understanding their weaknesses, the issues they face and where they work best in real situations. There will be special focus on the fact that traditional methods have subjective elements, the boundaries of current peanut assessment techniques and how advanced models can be useful in real-life situations.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 System Introduction

The system is designed to enhance peanut agriculture through image processing and deep learning techniques. The system is mainly focused on improving crop classification, disease, and pest identification. We utilized effective pre-trained models such as ConvNeXt (Base, Small, Tiny, Enhanced Tiny), EfficientNet (B0 and B4), and ResNet-50. This solution automates various tasks that are traditionally costly and require a large workforce. The main aim is to increase productivity, crop quality, and decision-making in peanut crops, especially in areas where technical expertise and real-time support are limited. This system plays a crucial role in providing sustainable agriculture and food security [1].

1.2 Background of the System

Traditional are different challenges in peanut cultivation, including unreliable manual type classification, late disease diagnosis, and ineffective pest management. Traditional agricultural practices rely heavily on expert knowledge, field experience, and manual labor, which causes subjectivity and inefficiencies. Previous research has explored machine learning and image processing techniques to address these issues, but many approaches suffer from limitations such as limited datasets, lack of scalability, and minimal real-world deployment. Our research fills these gaps by systematically reviewing the existing literature and developing an integrated solution using state-of-the-art deep learning models, focusing on practical implementation and usability for farmers.

1.3 Objectives of the System

The primary objectives of this system are:

- Enhancing the accuracy of peanut classification, disease detection, and pest identification using deep learning models.
- Develop a practical and user-friendly AI-based application for field use.
- Integrate multiple models for better decision-making in peanut agriculture.
- Evaluate model performance on real-world datasets and ensure computational efficiency.
- Review existing technologies and propose future research directions.

1.4 Significance of the System

This system has significant importance in the agricultural domain, especially for peanut farmers. It addresses real-world challenges such as crop loss due to late disease detection and inefficiencies in manual classification and pest identification. The system provides value by automating these processes using deep learning techniques and aids in reducing human error and optimizing crop management. Its integration with practical tools ensures that technological innovations are accessible and beneficial for the target users, particularly in resource-constrained rural environments. The system not only increases agricultural productivity but also contributes to food security, economic resilience, and sustainable farming practices. As peanut crops play an important role in enhancing the economies of several countries, which also enhance yield and crop health, they have a wide-reaching impact [2].

1.5 Scope of the project

This project focuses on three major aspects of peanut farming:

- **Classification** of peanut varieties.
- **Disease detection** to prevent crop loss.
- **Pest identification** to enable early intervention.

The project covers model training, evaluation, dataset analysis, and system design. Anyone can just upload a picture of the related issue, and it will help them in addressing these issues. However, it does not include hardware sensor deployment or end-to-end IoT integration. The target audience is agricultural researchers, farmers, and institutions involved in crop monitoring. The system will operate in environments with moderate computational infrastructure and internet access.

1.6 Challenges in existing solutions

Several limitations in current solutions highlight the need for this system.

- Lack of high-quality, labeled datasets for training and testing AI models. Creating these datasets is costly and requires domain-specific expertise [3].
- Manual classification and inconsistent pest evaluation methods.
- Overlapping symptoms in disease identification reduce accuracy.
- Inaccessible, expensive, or non-scalable pest detection systems.
- Limited user involvement in system design and lack of field-level adaptability.
- Many proposed methods are theoretical or experimental, not practical.
- Environmental and economic aspects are often overlooked in system design.

1.7 Key features of proposed system

The proposed system uses strong pre-trained deep learning models such as ConvNeXt-Tiny, Enhanced ConvNeXt, EfficientNet-B0, EfficientNet-B4, and ResNet-50 to perform various peanut crop-related tasks with high accuracy. It is designed with a modular architecture that allows for flexible implementation across key domains such as classification of peanut types, disease detection, and pest identification, all from image-based inputs. This system's development was grounded in a comprehensive evaluation of more than 58 peer-reviewed research papers, ensuring that it draws upon the most relevant and up-to-date academic insights. Moreover, the system is structured with scalability in mind, offering potential for deployment as a web-based application tailored for use by farmers and agricultural workers.

The design emphasizes real-world applicability, requiring minimal effort from users while delivering practical benefits in the field. To further enhance performance, the system employs hybrid and ensemble learning methods, which allow for more robust predictions and better adaptability to varying environmental conditions.

1.8 Expected outcomes

The expected outcomes of this project are both tangible and impactful. The system can produce highly accurate models for peanut classification, disease detection, and pest identification. These models are embedded within a simple interface that helps farmers to utilize the system effectively in real-time, without knowledge of technical things. As a result, the overall efficiency of peanut crop management is expected to improve, leading to better crop yields, reduced losses, and optimized resource use. Additionally, the system will support improved decision-making related to harvesting and disease control by providing timely, data-driven insights. Another key outcome will be the identification of existing dataset limitations, which will guide future data collection and research. Through this, the system not only addresses present agricultural challenges but also lays the groundwork for continued innovation in smart farming.

1.9 Research contribution

This project contributes significantly to both the academic and practical realms of smart agriculture. From an academic perspective, it offers a structured and thorough review of computational methods used in peanut farming, highlighting critical research gaps and suggesting promising directions for future inquiry. It also presents a comparative analysis of methodologies and performance metrics, adding clarity to the evolving landscape of AI applications in agriculture. Practically, the project delivers a functioning AI-based solution that incorporates multiple advanced models to monitor and manage peanut crops in a comprehensive manner. By addressing real-world challenges such as limited dataset availability, model interpretability, and usability in field conditions, this work bridges the gap between research innovation and real-world utility, enabling more sustainable and data-driven farming practices. Ultimately, it empowers farmers with modern tools while contributing to the broader goal of sustainable and intelligent agriculture.

2 REQUIREMENT SPECIFICATIONS

2.1 Product Scope

The proposed system is an AI-image-based analysis tool. This system is developed to help peanut farmers and researchers identification of diseases and pests and for the classification of peanut types. It utilized powerful deep learning models such as ConvNeXt (Tiny, Small and Base), Enhanced ConvNeXt Tiny, EfficientNet-B0/B4, and ResNet-50 to deliver accurate results using visual data. The purpose of this system is to bridge the gap between complex

Artificial Intelligence technologies and real-world agricultural applications by providing an impressive interface. It automates the processes which was performed manually in past. While the current system focuses on peanuts, it is modular and scalable to potentially support other crops in future iterations. However, it is limited by the quality and quantity of available datasets and environmental conditions that may affect performance in the field.

2.2 Functional Requirements

The functional requirements of the system are divided into the following core modules:

- **Image Upload Module**: Users can upload images of leaves, pods, or plants of peanut crops for analysis.
- **Peanut Type Classification**: The system classifies the uploaded peanut images into predefined categories based on visual characteristics.
- **Disease Detection**: Identifies common peanut leaf diseases, such as early and late leaf spot, using image classification techniques.
- **Pest Detection**: Detects signs of pest infestation based on image analysis, helping in guiding timely intervention.
- **Model Selection**: Allows switching between different pre-trained models such as EfficientNet-B0/B4, ConvNeXt variants, for performance comparison.
- **Result Visualization**: Displays results along with model confidence scores and recommendations.

2.3 Non-functional Requirements

2.3.1 Performance Requirements

The system should give results within 5–10 seconds for each image. The backend must support scalability to execute multiple requests simultaneously in a web-based version. Depending on the task and dataset, the accuracy for models should be above **85%**.

2.3.2 Safety and Security Requirements

All uploaded images must be stored securely with user consent and privacy protection in compliance with data protection regulations. The system should prevent unauthorized access by role-based access control if hosted online. Secure APIs and encrypted communication protocols (HTTPS) must be used for data transmission.

2.3.3 Software Quality Attributes

- **Maintainability**: The system has a modular architecture, and it allows easy updates or integration of new models.
- **Reliability**: The system should handle exceptions carefully and

maintain availability during user interaction.

- **Usability**: Intuitive UI to allow non-technical users, such as farmers, to interact with less training.
- **Portability**: The system should be deployable on local machines and adaptable to web-based platforms.

2.4 Data Requirements

- **Input Data**: Images of peanuts.
- **Output Data**: Predicted labels and scores of peanut type, disease, and pest.
- **Storage**: User information and result analysis will be securely stored in the database.
- **Data Formats**: The following formats, like `.jpg`, `.png`, and `.jpeg, are accepted`. Outputs are returned in JSON format for APIs or HTML-based display for web UI.
- **Compliance**: Data utilized in training should follow copyright and licensing terms.

2.5 Risk Analysis

The limited availability of peanut crop datasets may negatively impact model training. Transfer learning and data augmentation to expand existing data can be done to avoid this problem. Models may affect performance due to varying lighting, background, or camera quality. We can include diverse conditions in the training set for this problem. Farmers may hesitate to adopt AI tools. The solution is to develop a simple, guided UI and offer community education sessions. Transmission into web applications may cause performance constraints. However, optimizing models and code bases for lightweight deployment can be utilized

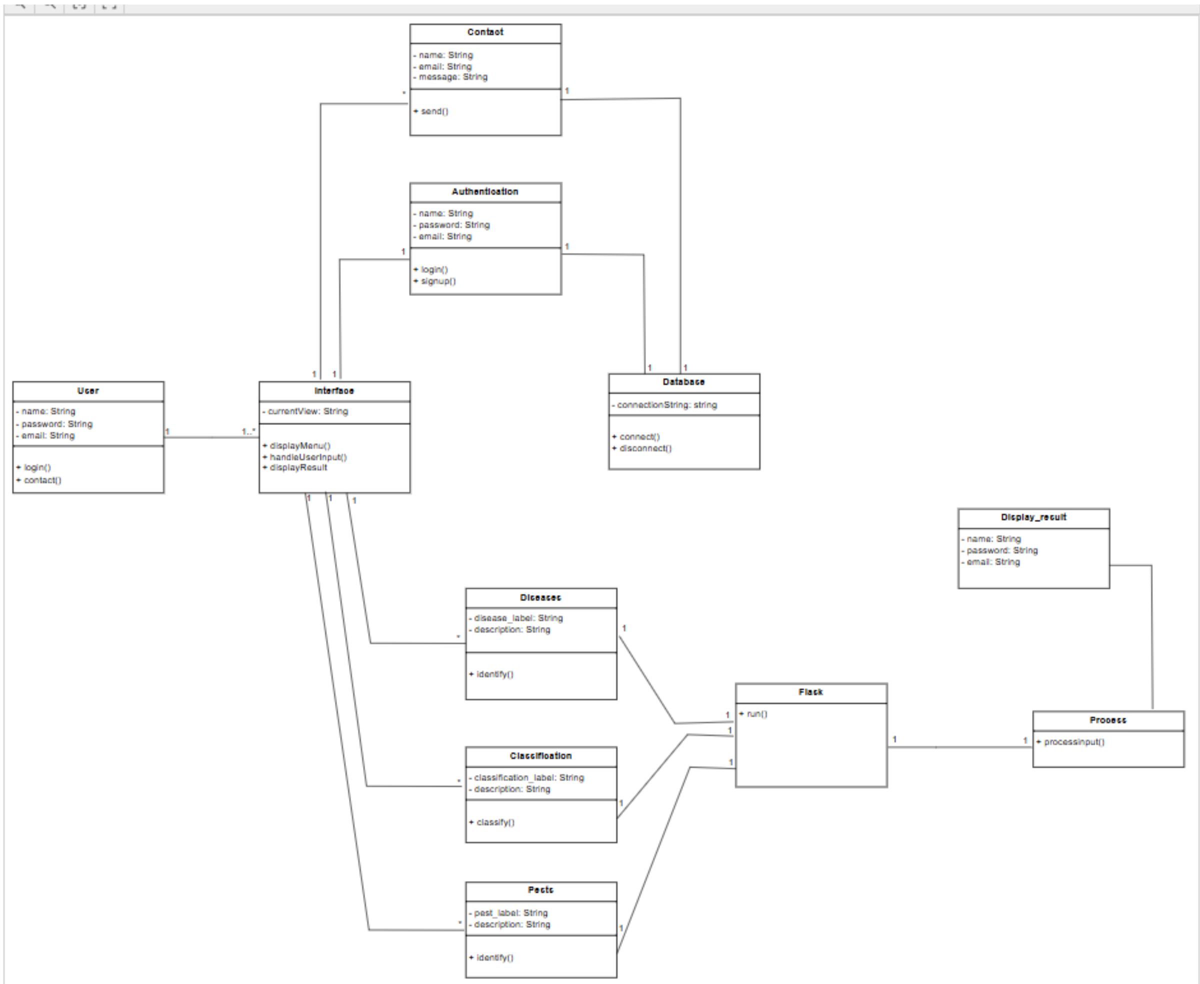
3 INPUT AND OUTPUT DESIGNS

3.1. Diagram:

To understand how a system works, diagrams are very helpful. They make it easy to see complicated processes, how they relate to one another and how they work together. The documentation contains several types of diagrams and each type is used for a specific reason. Class diagrams display how classes are organised and related by showing their attributes and methods. They illustrate the workflows happening within the system, by showing the major tasks, decisions and the order they occur. Use case diagrams display how the users (actors) use the system, labelling them and giving a purpose for each one. They show how objects are connected and communicate with each other in a particular scenario. Sequence diagrams show how objects communicate by passing messages to reach a specific goal. All the diagrams together give a clear picture of how the system is designed and how it works.

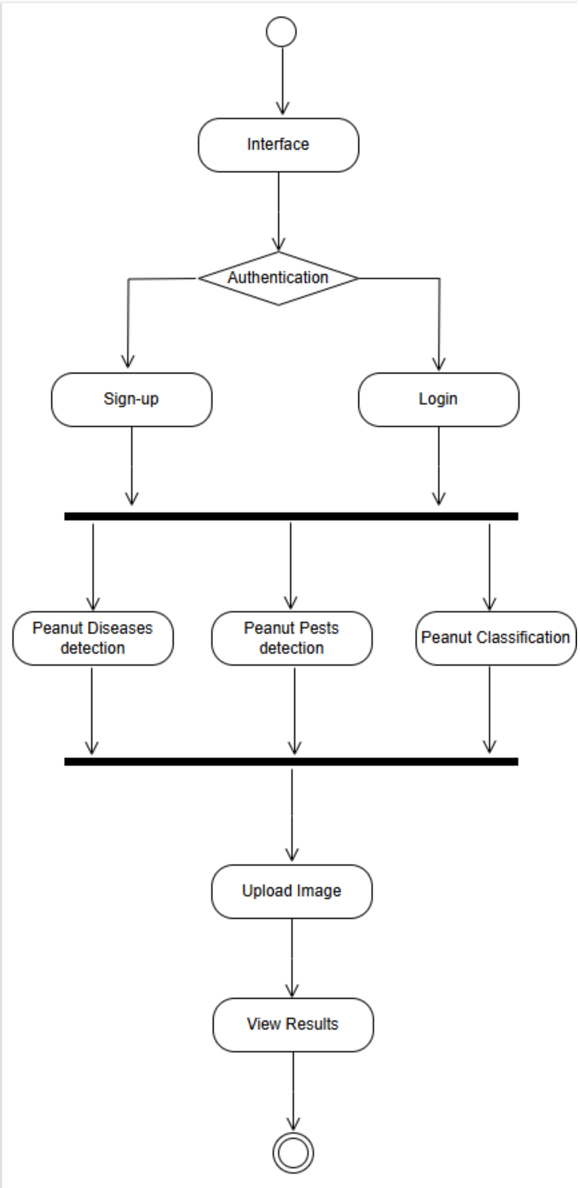
3.2. Class Diagram:

In object-oriented design, the class diagram helps to show the connections among classes, what their attributes are and which methods they have. Our method, the class diagram shows how the system is arranged and structured. Main classes are called User, Interface, Database, Pests, Classification, Diseases, Flask, Display_result Process and Contact. These classes talk to each other to bring about the system's functions. User class perform tasks including login and contact through the Interface and the Interface retrieves and stores data from the Database. Because these classes are linked this way, the system runs properly, helping users use data, process it and present results effectively.



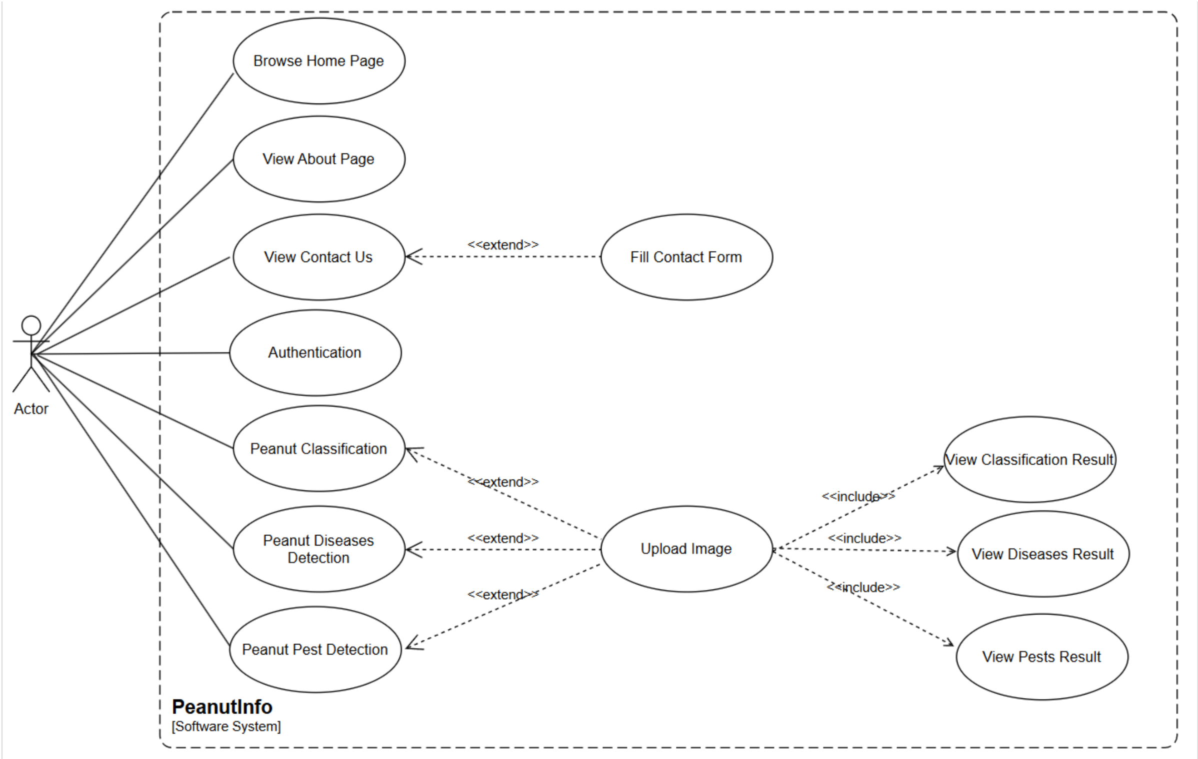
3.3. Activity Diagram:

The activity diagram illustrates the different workflows and activities in the system by highlighting important decisions and how they follow each other. It answers how processes move and the connections between different devices in a system. The process starts with the user getting access to the system and dealing with the interface. Some of the things a user is able to do are authentication, classification, disease detection and pest detection. Every activity starts with uploading pictures, then moves on to running data through software and finally shows the answers. Choices are made at different stages, for example checking if the user has passed authentication and if the uploaded image is valid. This diagram demonstrates clearly the flow of operations and the order in which activities are carried out in the system.



3.4. Use Case Diagram:

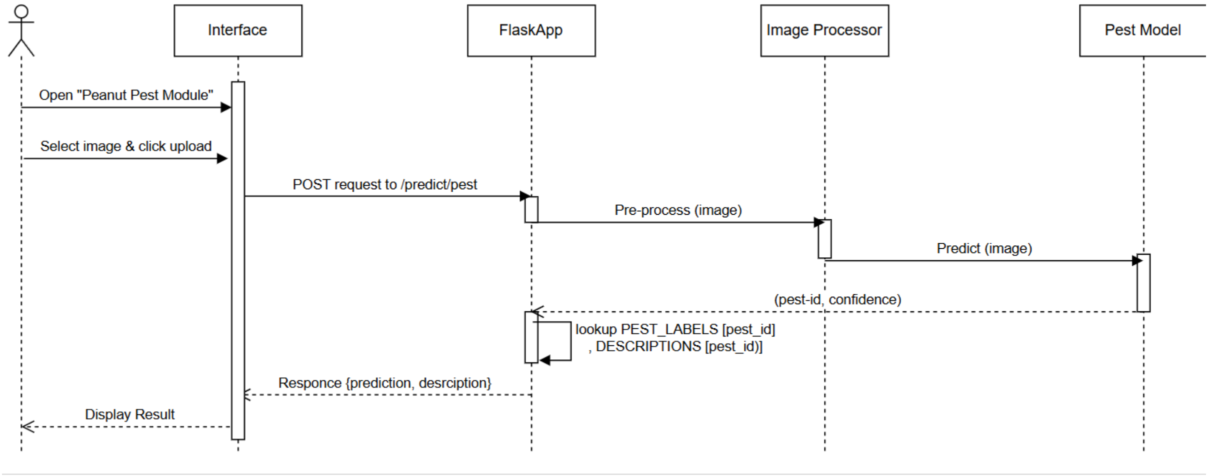
A use case diagram explains how users (the actors) use the system, clearly outlining what the system can do. It names every use case and tells what it does. The main person using the website, the user, has options like browsing the main page, hearing about the company, contacting administrators and conducting classifications of peanuts, disease screenings and pest identification. Every use case is a specific way the user can use the system. For example, the user authenticates themselves to use the features, fills out a message form to send a message, sends in an image for analysis and can look at the outcomes of the three processes. It shows how the user uses the system and the functions the system includes.



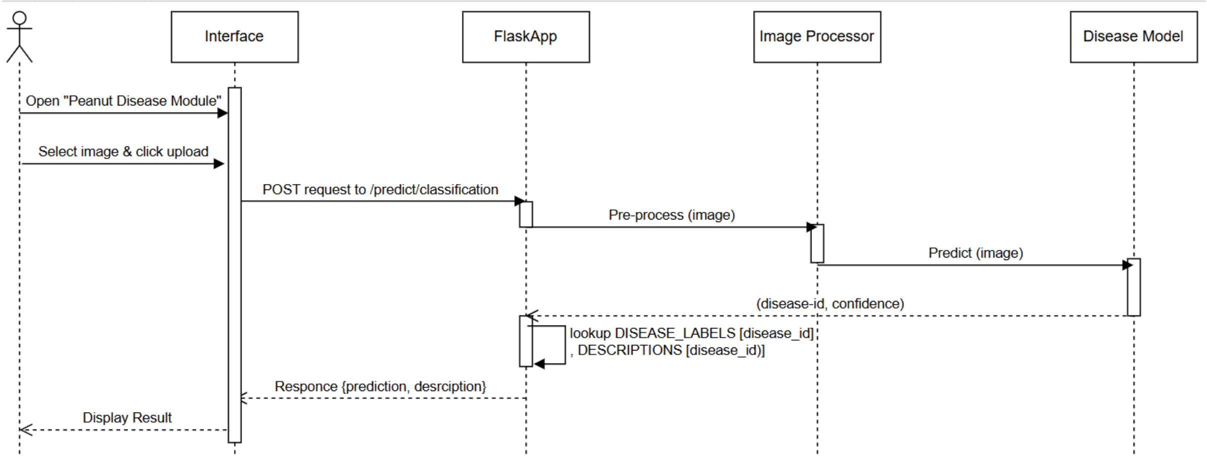
3.5. Sequence Diagram:

The diagram illustrates how objects send messages to one another to accomplish a function. It offers a methodical view of the ways things in physics reflect one another. As in the classification of peanuts, the user starts by choosing the classification option from the interface. The interface asks for necessary info from the database which sends the data back. The interface shows the data and the user selects an image to have its class determined. The interface passes the image to the classification class which works on it and sends back the result to the display result class. The display result class shows the result of the classification to the user. With this sequence diagram, you can easily understand how objects in the system communicate when peanuts are being classified.

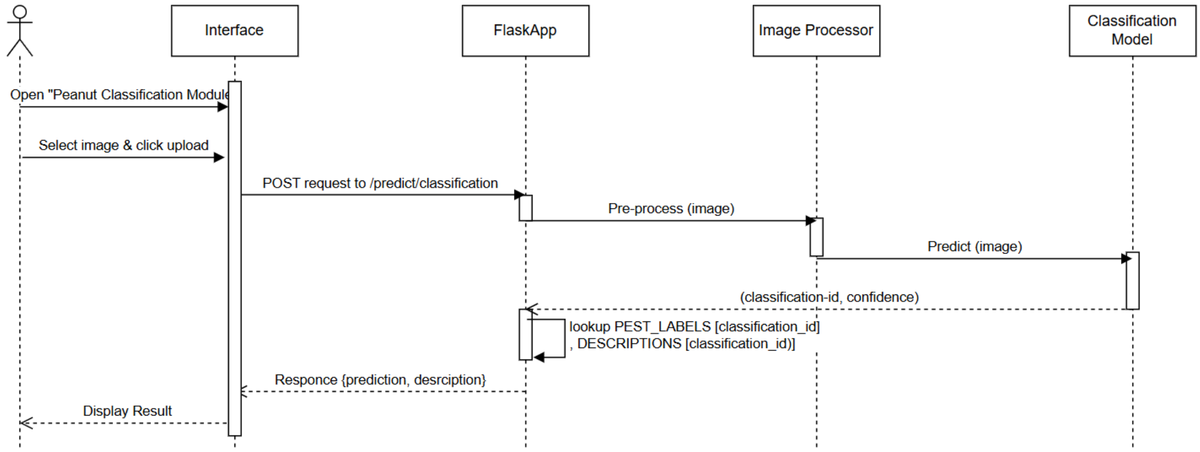
This is the Peanut pests sequence diagram;



This is the Peanut diseases sequence diagram;



This is the Peanut classification sequence diagram;



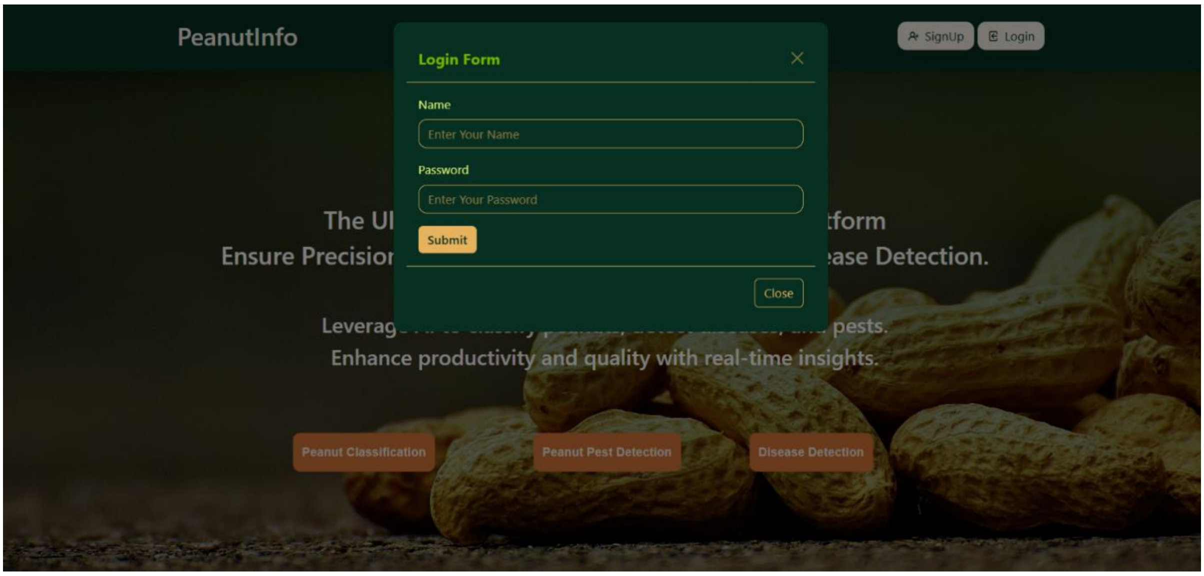
4 DESIGN SPECIFICATIONS

4.1. Introduction

The system holds a structured and modular design to support scalability, maintainability, and practical deployment in peanut agriculture. It is categorized into different modules for classification, disease, and pest detection. These modules utilized advanced pre-trained deep learning models. Modular design allows for independent updates and improvements without affecting the entire system, ensuring flexibility for future improvements. The design process used Python, deep learning frameworks such as PyTorch, and web technologies such as FLASK, HTML CSS, BOOTSTRAP, PHP, MYSQL to build a user-friendly interface that enables easy image upload and accurate predictions. Overall, the design ensures a practical, effective, and efficient solution for real-world peanut crop farming.

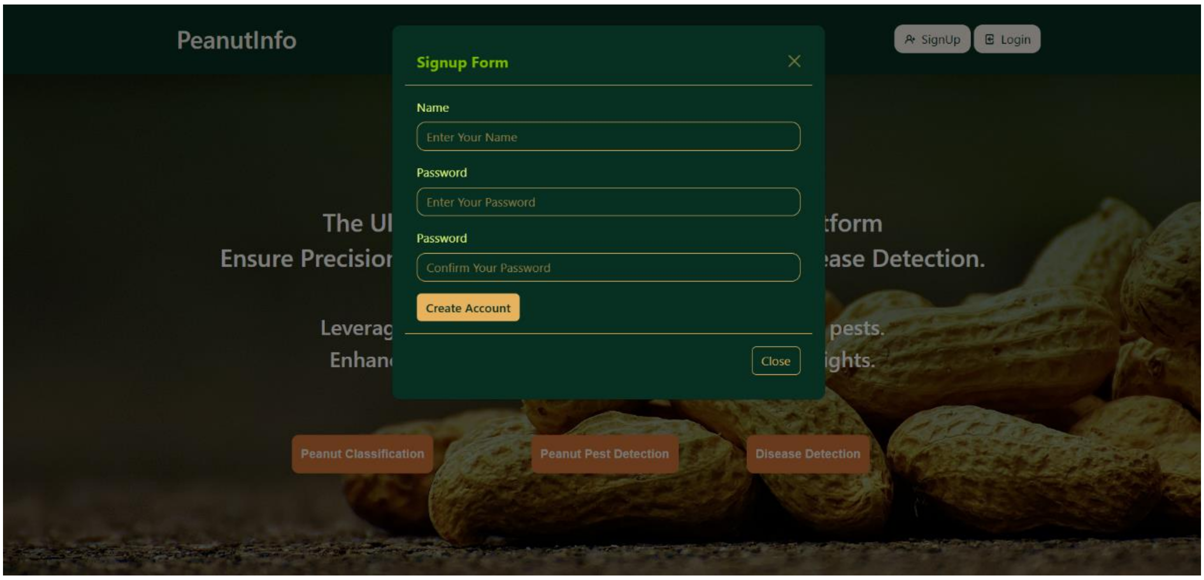
4.2. Login Interface

The login module provides secure access to the system's core functionalities. It has a simple and decent form where users can enter their credentials including name and password to authenticate their identity. It is made with simplicity in mind; it offers privacy to the users and restricts unauthorized access.



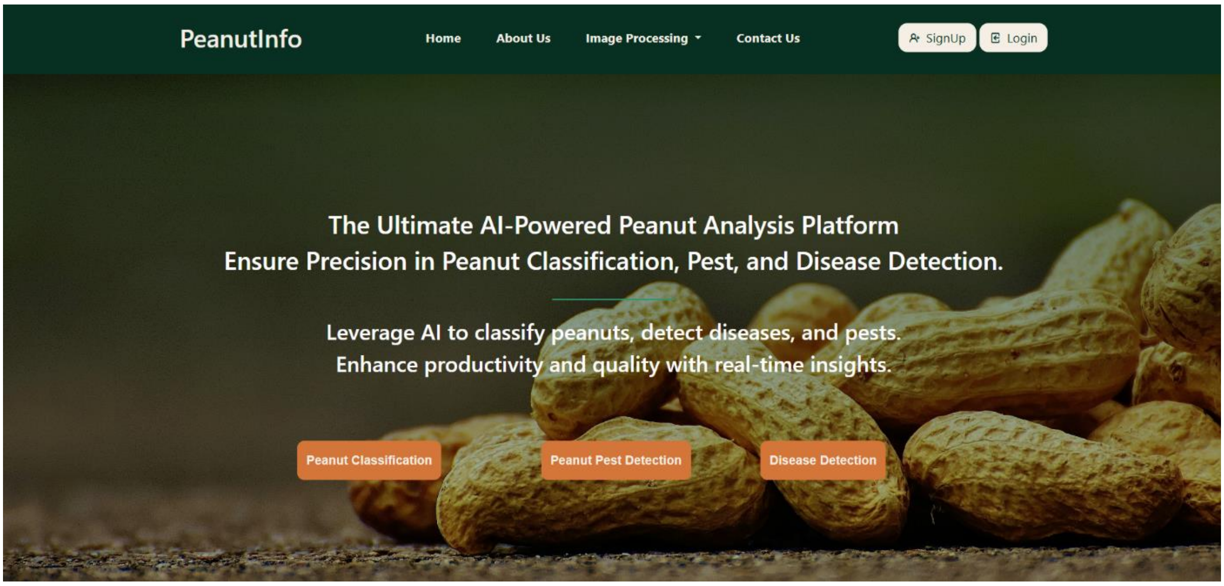
4.3. Signup Interface

The signup module provides secure access to the system's core functionalities. It has a clean, modal-style form where users can enter their credentials including name and password to create their own accounts and then use the model. It has very effective and efficient in design.



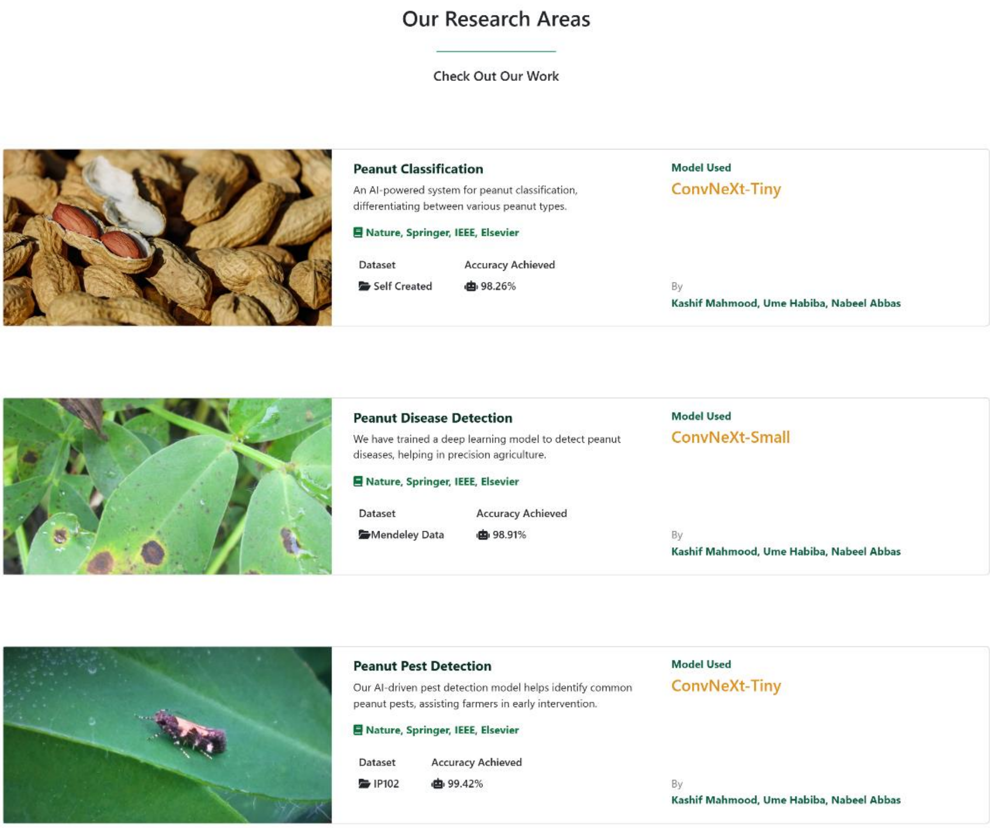
4.4. Landing Page – Hero Section

The landing page provides a gateway to the AI-powered peanut analysis platform. It highlights the system's core functionalities such as peanut classification, pest detection, and disease identification. It has a clean, user-friendly layout. There are separate buttons for each section where user can upload picture for analysis. This section provides real-time precision and productivity enhancement using Artificial Intelligence, aiding both farmers and researchers.



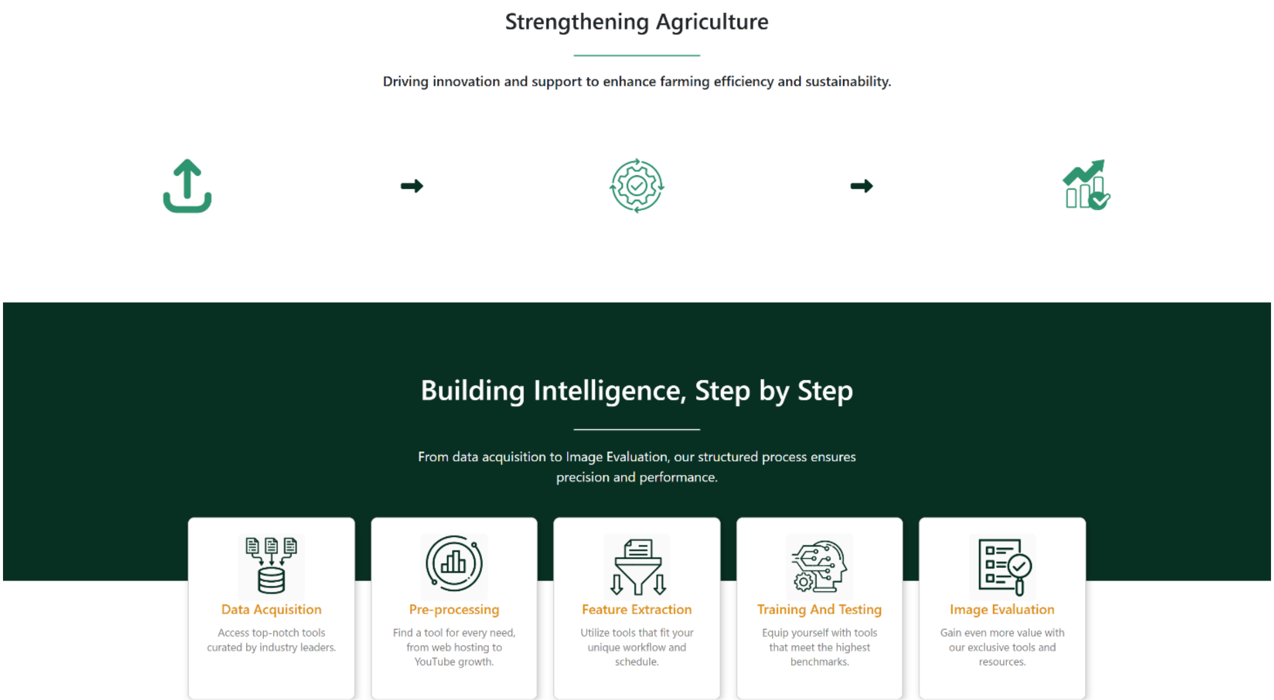
4.5. Research Areas Interface

This section shows the major modules of the system such as Peanut Classification, Disease Detection, and Pest Detection. Each module is explained with the model, dataset source, and achieved accuracy, providing users with clear insight into the technical backbone and effectiveness of each functionality. It provides the detail information of system.



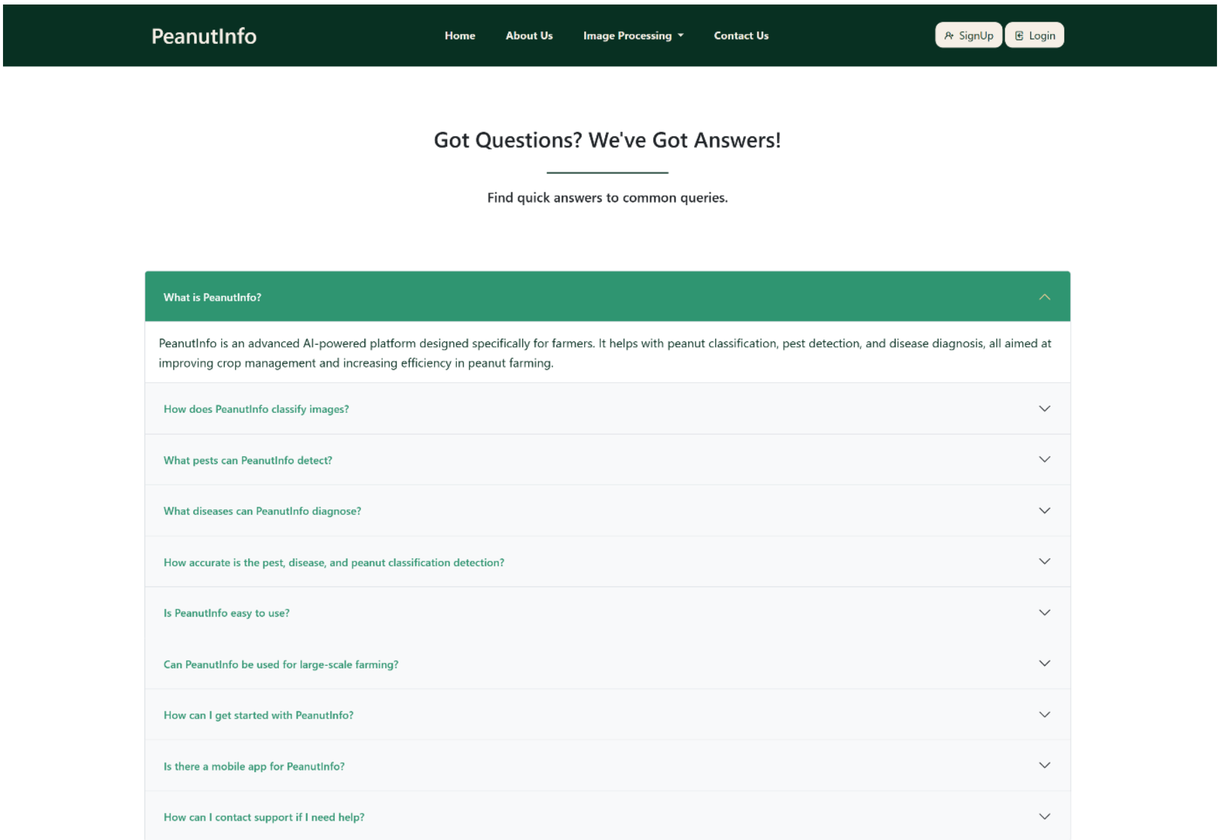
4.6. Process

This interface visually represents a structured workflow of our project work. It highlights a step-by-step pipeline including **Data Acquisition, Pre-processing, Feature Extraction, Training and Testing**, and finally **Image Evaluation.** Each phase is clearly illustrated with icons and concise descriptions, reflecting the systematic approach used to ensure precision, efficiency, and scalability in agricultural data analysis. This layout helps in user understanding of the technological process behind intelligent agricultural solutions.



4.7. Frequently Asked Questions

This interface displays the **FAQ (Frequently Asked Questions)** section of the *PeanutInfo* platform. The purpose is guiding users through its features and functionality. It provides concise, expandable answers to common queries such as how the system classifies images, detects pests and diseases, and supports peanut farming operations. The layout is simple and efficient which ensures that anyone can read it and get information easily, especially farmers. This improves accessibility and enhances user trust in the platform's AI-driven capabilities for agricultural decision-making. The FAQ section contributes significantly to user services and support.

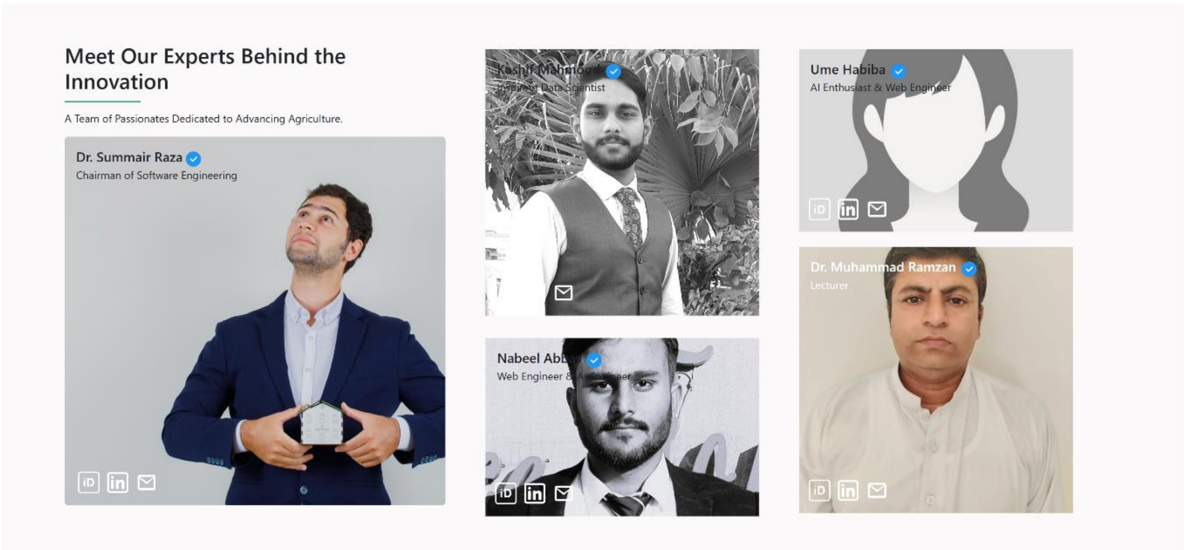


4.8. About Us

This "About Us" interface introduces the core mission and background of the PeanutInfo platform. The section highlights the platform as a result of a final year Software Engineering project, embedding technological expertise with agricultural needs. It also acknowledges the project team and mentor, which highlights the human effort behind the innovation. The layout, with an image of a farmer in the field, aligns with the platform's goal of enhancing farming through practical, tech-driven solutions.

4.9. Our Team

Nabeel Abbas, who is passionate about technological innovations; Kashif Mahmood, a dedicated data scientist; and Ume Habiba, who has a strong interest in image processing. The team is guided by our supervisors, Dr. Muhammad Ramzan and Dr. Muhammad Summair, whose mentorship and expertise have been instrumental in shaping this project.

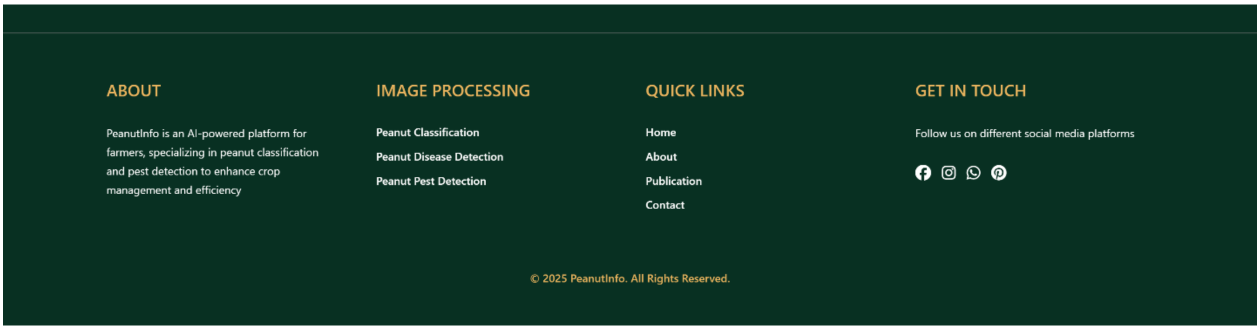


4.10. Followed Journals

In the development of our project, we extensively referred to high-impact research published in leading journals such as IEEE, Springer, Nature, and Elsevier. These sources provided valuable insights into advanced techniques in artificial intelligence, image processing, and agricultural technology. By following peer-reviewed literature, we ensured that our methodology is grounded in scientifically validated approaches. This helped us design a robust, reliable platform aligned with current global research standards. Leveraging such reputable sources has strengthened the credibility and technical foundation of our work.

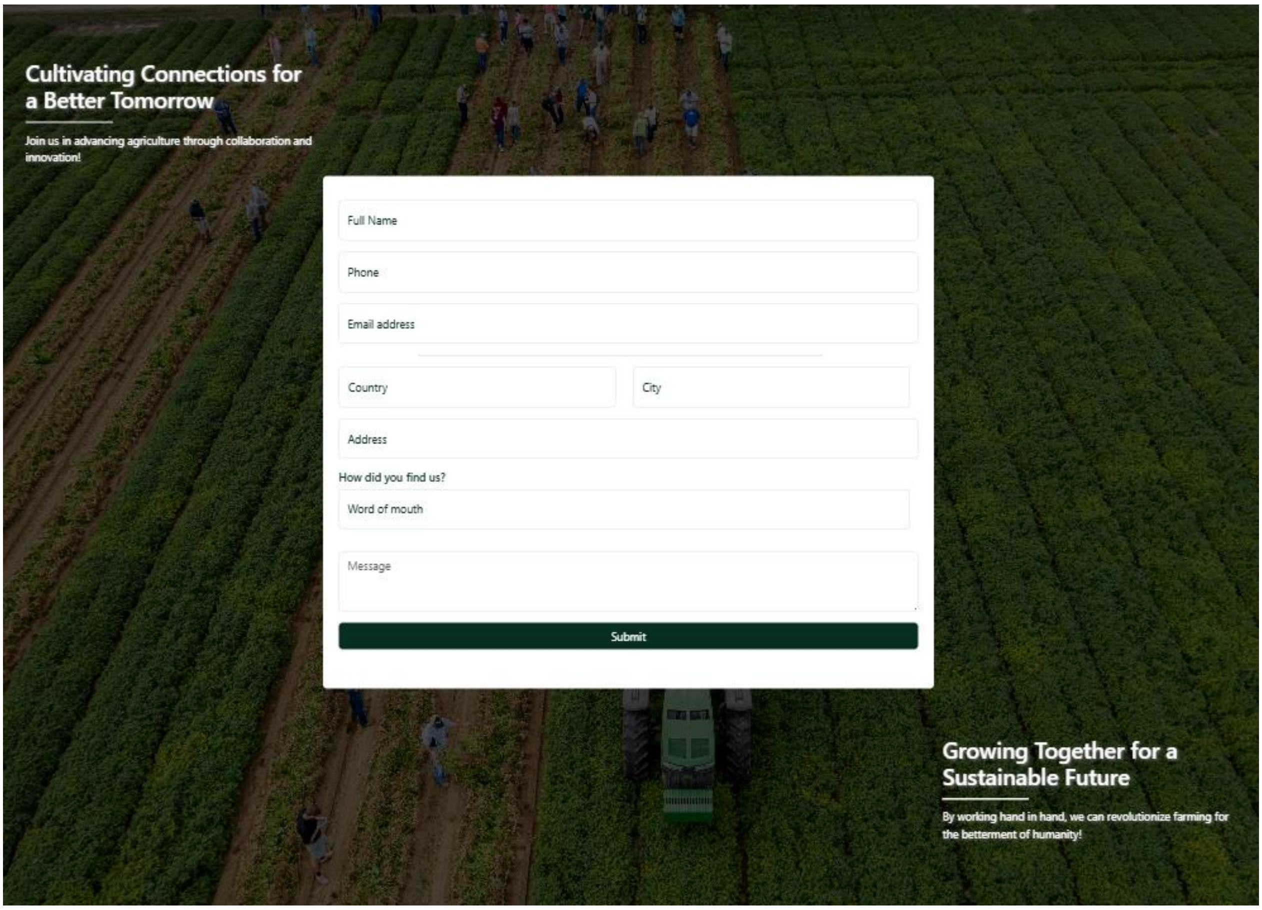
4.11. Footer

This includes all the necessary information such as about and provides necessary links for this site.



4.12. Contact Form

Here's a simple and well-designed contact form through anyone can easily contact with us by providing name, phone, and email address etc.



5 DEVELOPMENT AND TOOLS

5.1. Introduction:

This chapter presents the full methodology that was pursued as the project progressed through the development process. It talks about the measures and critical decisions that guided the development process from idea conception up to the creation of a usable prototype. An effective strategy, which was organised but flexible proved crucial in overcoming the peculiar challenges involved in the development of an AI powered application. All the vital tools and technologies are discussed in detail in this chapter. The intelligent selection and deployment of appropriate development instruments and software engineering techniques were very important. In this chapter we will present the development framework systematically.

5.2. Development Plan:

Firstly, different development phases were effectively planned in such a way which allowed for flexibility and continuous improvements. The first phase involved the detailed requirement analysis phase in which we collaborated with different domain experts to identify the necessary features. We did a comprehensive literature review on existing Peanut classification, disease, and pest detection techniques. Targeted users were defined properly. Next phase was focused mainly on the design part. We created few mock-ups for the web interface and created a database design structure that will facilitate instant storage and retrieval ease for user authentication records. Next phases involved the actual development of frontend and backend. After that AI models were trained properly. Frontend, backend, and AI models were integrated for the seamless performance.

5.3. Development Tools:

Development tools were selected depending on project demands, team experience and the availability of free, open-source software. For frontend, we have used several programming languages and frameworks like HTML, CSS, JavaScript, and Bootstrap. Html helps to create the basic structure of a website and CSS helps in its designing and choosing the best colour schemes, fonts, etc. JavaScript helps to add some functionality. Bootstrap is a popular CSS framework used for developing responsive and mobile-first websites quickly and provide some common components like forms, buttons etc. For backend, we've used PHP, MYSQL, and Flask (Python). Whenever, we'll give an image, a Flask API will be called and it is having all the trained models in it. It will be provided us the required results accordingly. PHP and MYSQL are used in the signup, login, and contact forms.

5.4. Conclusion and Future Work:

The development of the Peanut Disease, Peanut Pest detection, and Peanut classification system needed integrating various technologies to achieve the set objectives. To complete the system, a wide range of technologies were integrated, allowing the project to be accomplished. We ensured steady progress by splitting our work and following the principles of agile methodology. After combining all the parts, a functioning prototype was built that did what it was designed to do. The system achieves what it was created to do, but the process of building it has pointed out opportunities for advancements later on such as deployment of our model on cloud for better scalability and resource management. We can also improve the user experience by doing some changes according to the customers needs and feedback. They are designed to advance the current state by using improved technology and strategies to improve the system for everyone.

5.5. Version Control and Collaboration Tools

For this project, Git was used as a distributed version control tool. Git provides different functionalities making it a best choice. GitHub is available as a remote platform where we can set up our Git repositories. GitHub provides easily collaboration with other people. We kept in touch with WhatsApp and used email to inform the team, share issues, exchange ideas and plan our tasks. As a result of this method, the project's code was reliable, the team developed efficiently and the history of changes to the project was complete.

5.6. Software Engineering Practices

A strong system which can be maintained and upgraded in the future is possible only if software engineering practices are strictly followed. The team used an approach where development is done in steps and batches. The tasks necessary for development were divided in line with the overview from the development plan. Thanks to this approach, changes and modifications could be made with flexibility. Few changes in one module affected the others. The group wanted the design to be simple and easy for any user to manage, image upload and understand the results provided. There were a number of additional versions created for the project. Feelings from testing and demonstrations influenced how the software was improved as more cycles took place. The project used these strategies to achieve better results, faster coding and a simpler code that can be expanded further.

5.7. Security Measures and Tools

Developers made sure that the app and its user information were secure, as the app allowed both authentication and image uploads. We have used special functions like htmlspecialchars() so that values will be converted to special characters that will help to prevent the XSS attack and SQL injection. Passwords that users created were not saved in readable format. Instead, the database saves passwords that have already been hashed by the PHP function password_hash(). To verify that someone entered the correct password during login, the function password_verify() was used. Since every system has some risks, these security steps were added to guard final year project data and applications against common attacks.

5.8. Challenges and Mitigation Strategies

As the project was developed, we were aware of certain issues and actually faced them. Detecting peanuts with high accuracy, classifying their diseases and spotting pests is not simple because visual changes are tiny, images are not consistent in quality, brightness can vary and there is a limited amount of data. So, we handled all the data carefully, applied several pre-trained models on it, and evaluated the performance of model regularly. It was very hard to find a big and useful set of peanut images related to different categories, diseases and pests.

Checked if any current agricultural image collections were useful or could be utilized. We looked into getting images from the web while following all the rules outlined. Introduced a regular procedure for resizing and normalizing the data when feeding it to the models. It was necessary to make certain that HTML/CSS/JS, PHP for user management and Python/Flask for AI model serving all shared data and communication properly. So, we tried to keep each step of development simple and flexible. Developing solutions for these concerns depended on using technology, planning well, being flexible and consistently solving problems. It was essential to use these mitigation strategies for the project to clear all the given restrictions.

6 QUALITY ASSURANCE

6.1. Introduction

QA plays a vital role in the way we create our project. For the Peanut Detection system, QA requires checking every part and adjusting them to ensure the best results are achieved. A QA plan that supports the project's needs is the first step in the process. Here, I outline what we will test, our main testing methods and the tools we will require for this acceptance testing.

6.2. Testing strategies and methodologies

A range of testing is implemented within our project. We conduct unit tests to check all the parts separately such as the image preprocessor and the AI model. The next process in testing ensures the frontend and backend systems communicate smoothly. At this step, the application is checked against the given requirements. To check if the system works well in practice, farmers and specialists from the agriculture field undergo acceptance testing.

6.3. Automated Testing

Using automated testing. We rely on automated testing to help us manage the complexity of our application that uses AI. For frontend testing, we use Selenium to write scripts that make the browser act like a user does. Testing the APIs and database in the backend is done with pytest. We run the automated tests each night so that we soon discover and fix any new issues or errors in our code.

6.4. Performance Testing

Testing the System's Performance Processing large collections of images works well only when the AI performs well. We run JMeter to generate user traffic, monitor the server's speed and load capacity and check the resources being used. As a result of these tests, the system works optimally, ensuring it continues to run smoothly and rapidly whenever we encounter peak demand, particularly in key seasons when farmers rely on it the most.

CONCLUSION

7.1. Summary

We have proven that AI can be used in practice to help farmers solve their agricultural challenges. The use of AI in peanut farming is now a major point of success with the help of this project. Our system is able to detect disease in peanuts, categorize different types of groundnuts and identify different pests from photos. Ensuring both the frontend and backend are responsive allows farmers and clients to access the information they need without difficulty. As well as the IT development itself, we also contribute to both the theoretical and practical realms. We have developed a comprehensive review about detecting peanut diseases, differentiating peanuts and identifying peanut pests. We have also written two technical papers: one for peanut diseases and another for peanut pests. They review the techniques, barriers and original solutions that we have established. Our testing assures that our AI models are always accurate and consistent. Moreover, their content enhances agricultural AI research and encourages others in the field to improve their work. The solution we provide to farmers meets the major challenges faced in peanut farming. As a result of this project, we have designed a valuable system that helps to correctly classify peanut images, identify any diseases present and expose pests. The web application works very smoothly, thanks to using HTML, CSS, JavaScript, Bootstrap for the frontend and PHP, Flask API for the backend. Thanks to the diversity in the data we trained on, our AI models have proven highly accurate in helping farmers manage their crops. The project has already reached its first aims and may make a big difference in the area of agriculture. Thanks to AI, farmers have the information needed to help their crops thrive and avoid damages. Since it keeps adapting, the system remains both useful and relevant for the long run. While building it, we had to deal with issues in improving the model and making the system faster for handling several requests. Yet, they led us to step up and improve data preparation steps and build cache structures. The things I have learned so far will certainly help me in any future tasks. In the near future, it will be possible to make the system even more advanced in several ways. Trying to improve the accuracy of our AI models, we plan to enrich the database of diseases and pests. Letting sensors provide real-time information about the environment will greatly benefit farmers and this is now being considered. Moreover, creating a user interface according to feedback received by users will guarantee all users find it easy to use. In all, this project has been filled with new ideas and successful explorations. We have put a system in place that deals with present challenges and also supports advancements in the future. We are looking forward to watching the tool improve and benefit the farmers.

8. References

[1] A. Khan, A. Nawaz, and F. Ahmad, "Artificial Intelligence in Agriculture: A Review," *Int. Res. J. Modernization Eng. Technol. Sci.*, vol. 6, no. 3, pp. 4358–4363, Mar. 2024.

[2] FAO, *The State of Food Security and Nutrition in the World 2022*. Rome, Italy: Food and Agriculture Organization of the United Nations, 2022.

[3] G. Wang, Y. Sun, and J. Wang, “Deep learning-based plant disease detection: A comprehensive review,” *IEEE Access*, vol. 8, pp. 3646–3657, 2020, doi: 10.1109/ACCESS.2020.2963957.

IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.Doi Number

A Comprehensive Review of Existing Techniques for Peanut Type Classification, Diseases, Maturity, and Pest Detection

Ume Habiba[1], Qaisar Abbas[2], Nabeel Abbas[1], Turki Alghamdi[2], Kashif Mahmood[1], Sami S. Albouq[2], Muhammad Ramzan[1,*], Saad Said Alqahtany[2]

[1]Department of Software Engineering, University of Sargodha, Sargodha 40100 Pakistan

[2]Faculty of Computer and Information Systems Islamic University of Madinah, Madinah 42351, Saudi Arabia

Corresponding author: Muhammad Ramzan (email: mramzansf@gmail.com)

ABSTRACT Agriculture serves as the backbone of many economies, providing food security and contributing significantly to national income. Among various crops, peanuts hold substantial importance as a financial crop, and oilseeds directly impact the economy of the country. Our comprehensive review aims to analyze and evaluate existing computational approaches in peanut agriculture across four critical domains: classification, maturity assessment, disease detection, and pest detection. The primary focus of this research is to identify current technological gaps and propose future research directions to develop more efficient, cost-effective, and practical solutions for managing peanut crops. As there are various types of peanuts, it is compulsory to identify them because they influence productivity and quality. We seek to systematically assess the current computational approaches, with a focus on their limitations, challenges, and applicability to real-world scenarios. Special attention will be directed towards the subjectivity inherent in traditional approaches, the limitations of current maturity assessment techniques and the practicality of sophisticated computational models in the real world. Moreover, peanut leaf diseases and pest infestations are major risks to crop health and productivity, directly affecting yield and quality. Disease and pest detection systems often face objections in real-world scenarios due to environmental variables and complicated field conditions. In this review, we finally selected approximately 58 research papers, classifying them into four sections: classification, maturity assessment, disease detection, and pest detection. For each section, we examined the accuracy, datasets, methods, and techniques used, organising them based on deep learning, machine learning, and hybrid approaches. Each section presents an in-depth description of the data and findings related to its respective area of research, highlighting both achievements and areas requiring further investigation.

INDEX TERMS Maturity, Diseases, Classification, Pests, Machine Learning, Deep Learning, AI, CNN

I. INTRODUCTION

Agriculture is considered an indispensable pillar of a country's social and economic development, serving as the foundation for addressing the basic need for food for human life. Rapidly increasing demand for grain consumption worldwide highlights the demand for sustainable agriculture to fulfil future needs. Despite the essential role of agriculture, 37.43% of the entire land region was utilized for agricultural production during 2000–2016 while the world population increased by 1.31 billion (World Bank, 2016)[1]. This invites critical challenges of producing sufficient food for the rapidly growing population and preserving environmental sustainability. The solution lies in adopting advanced technologies and improved agricultural practices with global collaboration.

Although agriculture has significantly evolved with the help of technology, striving to keep the balance between the demand and the contribution. The scarcity of assets in the field of agriculture attracted the attention of researchers and insisted on utilizing technology to enhance creativity to fulfil future demands. Technologies, in their various forms such as Data Science, and Artificial intelligence (AI) are spearheading a dramatic change in every field of life including healthcare, telecommunications, and the

1automotive industry with agriculture being no exception. The integration of AI with different technologies has unlocked new opportunities to explore, analyze, and gain valuable insights, clearing the way for more effective agricultural exercises. Extensive research and development have been conducted in the area of image processing, specifically concentrating on the applications related to peanuts including peanut classification, pest detection, disease detection, and maturity assessment. Image processing as a sophisticated machine learning collection method that encompasses biochemical processes and molecular processes, is a new plant DUS testing mechanism, which is sanctioned by the International Union for the protection of new kinds of plants (UPOV) [2]. The image processing technology is used to gather the appearance of seeds, and then recognize the different varieties based on these features [3].

Among various crops, peanut is a significant oil crop across the globe [2], also the backbone of many economies, playing a crucial role in ensuring food security and contributing significantly to national income. However, peanut cultivation faces several challenges, including the need for accurate identification of different peanut types, which directly impacts productivity and quality. Conventional methods of identification stand on manual trait analysis, which is often subjective, labor-intensive, and lacks scalability. Similarly, assessing peanut maturity using conventional techniques is both time-consuming and not trustworthy, leading to inefficiencies in crop management. Moreover, late and early leaf spot diseases identification, respectively caused by Nothopassalora personata and Passalora arachidicola, are devastating diseases of peanut (Arachis hypogaea) capable of defoliating canopies and plunging yield [3], such types of diseases and also pest infestations pose significant threats, reducing crop health, yield, and quality. To comprehend the scope of these challenges, we conducted a comprehensive analysis across multiple research platforms.

This review seeks to address these challenges by offering a thorough examination of the current evolution in peanut classification, maturity assessment, disease detection, and pest detection. Our contribution lies in identifying the shortcomings and challenges of existing approaches while illustrating the potential of integrating advanced technologies like deep learning, machine learning, and hybrid models. Despite their promise, these approaches face momentous challenges, including high processing cost, limited scalability, and variability in environmental factors impacting the performance of the model. For instance, high computing needs bound to the real-world pertinent of these models in resource-constrained environments. Peanut pods grow underground, which is the major specification of peanuts [4]. A significant challenge in peanut farming is the inability to precisely impose the maturity of underground peanut pods, which significantly hampers harvesting decisions and overall productivity. By systematically analyzing these confines, we focus on proposing practical and scalable solutions to address the specific needs of peanut farming. This study not only provides an extensive cutting-edge review but also identifies the areas for improvement and potential for upcoming studies and technological innovation.

We reviewed several research papers, categorizing them based on their methodologies, accuracy, datasets, and technological approaches. The paper is classified into four main sections to ensure clarity and systematic exploration of the subject matter. The first section focuses on peanut classification, outlining numerous methods and their effectiveness in distinguishing between peanut types to boost productivity and quality. The second section explored the maturity assessment techniques, critically evaluating their reliability and performance across varied datasets. In the third section, we investigate advancements in disease detection, emphasizing models that employ image processing and artificial intelligence for early identification and mitigation of crop diseases. Finally, the fourth section focuses on pest detection methods, analyzing their role in safeguarding peanut crops and ensuring sustainable yield. Each section provides a thorough discussion of the datasets, algorithms, and technologies employed, providing readers with a comprehensive overview of current trends and future directions in peanut-related research. This methodical approach facilitates a deeper insight into the challenges and innovations in the field, providing a roadmap for tackling critical agricultural issues. These findings highlight the critical need for innovative, reliable, and efficient solutions in peanut farming, aided by advancements in machine learning and artificial intelligence.

II. RESEARCH METHODOLOGY

This systematic review paper aims to find out the most effective approaches, tools, datasets and performance measure for peanut classification, disease detection, maturity assessment, and peanut pest identification. The systematic literature reviews helped us to understand the prior work in the field. All the data gathered from primary studies is categorized.

A. RESEARCH OBJECTIVES

This research was designed to address the following key objectives:

- Systematically review and analyze current methodologies for peanut classification, maturity assessment, disease detection, and pest detection.
- Identify gaps and limitations in the existing approaches to highlight areas that require further investigation.
- Suggest potential research directions and improvements to develop more efficient, cost-

effective, and practical solutions for peanut crop management.

- Explore ways to enhance the practical field implementation of AI models for peanut agriculture, considering computational efficiency and real-world applicability.
- Explored the benchmark datasets in this domain for peanut type classification, maturity analysis, disease detection, and pest attack identification.

B. RESEARCH QUESTIONS

This systematic literature review addresses the following questions:

- RQ1: What are the current methodologies used for peanut classification, and how effective are they in real-world applications?
- RQ2: What techniques are employed for peanut maturity assessment, and what are their limitations regarding accuracy and reliability?
- RQ3: How peanut diseases are being detected currently, and what challenges do they face in varying environmental conditions?
- RQ4: What approaches are used for peanut pest detection, and how can their efficiency be improved for practical field use?
- RQ5: What are the common datasets, methods, and accuracy rates reported in the literature for peanut classification, maturity, disease, and pest detection?
- RQ6: What are the identified technological gaps in the existing literature, and what future research directions can address these gaps?

C. SEARCH STRING

The search strings are used to accumulate an extensive set of relevant literature. Multiple academic databases like IEEE, Springer, MDPI, Elsevier, etc. are used to get relatable scientific content. The strings used for the search paper with the journal have been discussed in Table 1.

TABLE 1
SEARCH STRING FOR PAPERS

Journals	Search String	Context
IEEE, Springer, Elsevier Science Direct, MDPI, Nature, Teach science	("Peanut classification" OR "groundnut classification" OR "peanut type identification") AND ("artificial intelligence" OR "machine learning" OR "deep learning" OR "AI models") ("Peanut maturity assessment" OR "groundnut maturity assessment" OR "maturity detection in peanuts") ("Peanut disease detection" OR "groundnut disease detection" OR "disease identification in peanuts") ("Peanut pest detection" OR "groundnut pest detection" OR "pest identification in peanuts") ("Peanut agriculture" OR "groundnut agriculture" OR "peanut crop management")	Agriculture

D. SCREENING OF RELEVANT PAPERS

All research papers that we got from the search were not closely related to our research, so they needed to be screened according to their relevancy. According to the searching technique defined by Dybå and Dingsøyr [5] for selecting related papers, first, we excluded the documents based on the relevant title. For example, in our initial search, we got some papers that were related to other nuts so we excluded them and only focused on papers related to groundnut or peanut. Furthermore, we screened articles by reading their abstract and knowing the relevance of the paper for us.

The following types of paper were excluded:

- Articles that were not relevant to our field.
- Articles not according to the search string.
- Articles not having AI models.

E. STUDY SELECTION PROCESS

Our study consists of four sections: Peanut Classification, Peanut Maturity, Peanut Diseases, and Peanut Pests. Initially, when we applied a simple search process we got 640 papers on peanut classification, 623 papers on peanut maturity, 555 papers on peanut diseases, and 438 papers on peanut pests. An extensive screening process was applied to get the closely relevant papers only. The overview of the final selected papers has been given in Table 2.

TABLE 2
PRIMARY SELECTION PROCESS FOR RETRIEVED ARTICLES

Selected Papers for Peanut Classification

Stage	Method	Selection Technique	IEEE	Science Direct	Springer	IGI Global/ Frontier	MDPI/IFAC	Total
1	Search	Keywords	200	250	100	50	40	640
2	Screening	Title	120	100	80	20	16	336
3	Screening	Duplication Removal	80	60	50	10	10	210
4	Screening	Abstract	40	30	25	5	5	105
5	Screening	Full Article	10	8	6	2	2	28
6	Final Selection	Based on relevance and quality	4	3	3	1	2	13

Selected Papers for Peanut Maturity

Stage	Method	Selection Technique	IEEE	Science Direct	Springer	IGI Global/ Computers and Electronics in Agriculture	MDPI	Total
1	Search	Keywords	262	177	140	12	32	623
2	Screening	Title	27	13	10	3	2	54
3	Screening	Duplication Removal	19	9	8	2	1	38
4	Screening	Abstract	12	5	5	2	1	24
5	Final Selection	Full Article	4	3	3	2	1	13

Selected Papers for Peanut Diseases

Stage	Method	Selection Technique	IEEE	Science Direct	Springer	IGI Global	MDPI	Total
1	Search	Keywords	200	150	170	15	20	555
2	Screening	Title	30	26	20	1	2	80
3	Screening	Duplication Removal	21	19	14	1	1	56
4	Screening	Abstract	13	12	9	1	1	36
5	Final Selection	Full Article	4	5	4	0	1	14

Selected Papers for Peanut Pests

Stage	Method	Selection Technique	IEEE	Science Direct	Springer	IGI Global/ PLOS ONE	MDPI	Total
1	Search	Keywords	120	180	120	12	6	438
2	Screening	Title	17	14	9	1	0	41
3	Screening	Duplication Removal	12	10	6	1	0	29
4	Screening	Abstract	7	6	5	1	0	18
5	Final Selection	Full Article	5	2	2	1	0	10

F. FLOW OF ACTIVITIES

In this section, we delve into the sequence of activities essential for the development of a reliable groundnut image evaluation model. From the acquisition of high-quality groundnut images and preprocessing them for consistency to extract features, training, testing, and final evaluation of the image, each step contributes to the overall reliability and robustness of the model, as elaborated below.

1) DATA ACQUISITION

Defined peanut images are acquired in this foundational step. The dataset is typically sourced from agricultural farms or publicly available platforms. Datasets are labeled with different categories, providing a basis for model training. Data should be acquired carefully because some challenges like noisy data require robust strategies for data collection.

2) PRE-PROCESSING

Image pre-processing is done to enhance the standard of peanut images and eradicate the unpleasant commotion in the images. Raw images are examined. In this step, images are normalized, resized, and enlarged to ensure uniformity and improve the model's performance and validity. This step ultimately ameliorates the generalizability of the model.

3) FEATURE EXTRACTION

Feature extraction technique entails identifying the relevant characteristics or attributes or trends in the data. This step diminishes the dimensionality focuses on the most relevant data and improves the sufficiency of training and evaluation data.

4) TRAINING AND TESTING

This stage involves splitting or dividing of dataset into training and testing folders to analyze the model performance. Dividing the dataset into training and testing is crucial for developing trustworthy models. The conventional practice of data splitting is generally 70% or 80% for training the data and 30% or 20% for analyzing the data respectively. This ensures the model learns the patterns during training and is evaluated on unseen data.

5) IMAGE EVALUATION

In this step, images are evaluated based on their specific characteristics. Using a specific model, images are distinguished. Specific algorithms of a given model are applied to get the required output. All the images related to a specific module like Peanut Diseases, Peanut Pests, Peanut Classification, or Peanut Maturity are evaluated correctly by the model. The generic steps are shown in Figure 1.

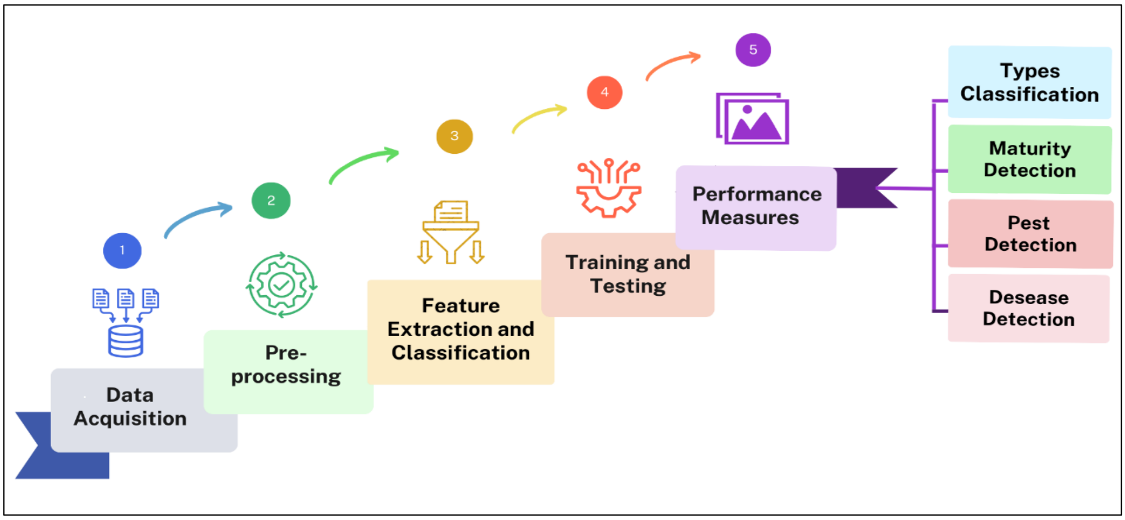


FIGURE 1. Basic steps for data Evaluation Model Development.

III. REVIEW OF EXISTING PAPERS ON PEANUT

In this section, the outcomings of the Peanut classification, disease detection, maturity assessment, and Peanut Pest identification are addressed as presented in Table 2. After the selection of the screening procedure, to create a positive response to the agriculture subject, the research investigations have been used to demonstrate the answers related to the specific research-related questions

A. PEANUT DISEASE DETECTION

In this section, we analyze existing studies on peanut disease detection, categorized into three main approaches: deep learning, machine learning, and hybrid models. Some sample images of peanut diseases are shown in Figure 2.

Tikka Early Leaf Spot Pestalotiopsis Phyllosticta

FIGURE 2 Visual representation of prevalent peanut diseases.

1) EXISTING RESEARCH WORK ON PEANUT DISEASE DETECTION USING DEEP LEARNING MODEL

Paramanandham et al [6] used the convolutional neural networks (CNNs) model for the detection of groundnut leaf disorders. In their paper, researchers collected a dataset of 10,361 images labelled with different peanut leaf images. Using the CNN model, they achieved an accuracy of about 97.225 %.

Maheswaran et al [7] used the convolutional neural networks (CNNs) model for the classification and spotting of groundnut leaf diseases. Researchers used the dataset of peanut diseases collected from different farms in their paper. Using the CNN model, they achieved an accuracy of about 96.50 %.

Vaishnnave et al [8] used the deep convolutional neural network model for the detection and authentication of peanut leaf diseases. Researchers used ten kinds of groundnut disease images that contain 6400 image samples from the Self-created dataset for training, and testing with the proposed DCNN model which showed better performance. They achieved an overall accuracy of 99.88% in each class.

Sasmal et al [9] used deep learning techniques especially convolutional neural networks (CNNs) have proved significant ability for early discovery of plant leaf diseases. In their paper, researchers collected a dataset of 1720 images that can be used to train the DL models for the early-stage detection of peanut diseases. The highest accuracy achieved in their paper is 96.51%.

Gebremeskel et al [10] used the combined deep learning defect classification model that integrates VGG16 and InceptionV3 for the defect detection of groundnut seed. In their paper, researchers collected a dataset of 7800 images. They divided the dataset into six classes healthy, virus defect, fungal defect, physical damage, and pest defect. They achieved an accuracy of 96.25%.

Kukadiya et al [11] deployed the Deep Convolution Neural Network (CNN) model 'CNN8GN' for the effective management of groundnut crops through the early prediction of leaf diseases. In their paper, researchers collected a dataset of total 5322 authentic images representing six distinct classes of Groundnut leaf diseases, captured from the agricultural fields in the state of Gujarat, India. They acquired a testing accuracy of 91.25%.

Sree et al [12] explored the progression of a Convolution Neural Network (CNN) model for discovering groundnut leaf diseases using a mobile net. The "Mobile Net CNN Plant Disease Detector" is a deep learning model that uses images of groundnut leaves to identify numerous diseases using data augmentation.

Kaur et al [13] employed fine-tuned Inception V3 model for the multiclass classification of groundnut disease. The method utilizes plant leaf images to discriminate between healthy and unhealthy leaves. The Inception V3 model is used to identify and divide plant diseases in groundnut crops, proving more efficient than traditional approaches and demonstrating the potential of deep learning in agricultural management. In Table 3 existing work for peanut diseases using deep learning has been discussed.

TABLE 3
REVIEW OF EXISTING MODELS FOR PEANUT DISEASE IDENTIFICATION USING DEEP LEARNING

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[6]	2024	Use of LeafNet for groundnut leaf diseases	10,361 images	CNN model	97.225%	IEEE
[7]	2022	Identifying and Classifying Groundnut Leaf Disease	Data sets collected from farms	CNN model	96.50%	Springer, Cham
[9]	2024	Groundnut leaf dataset for groundnut leaf diseases	1720 original images	Deep learning (DL) techniques, specifically CNNs	96.51%	Elsevier
[8]	2020	Method for classification of groundnut diseases	6400 groundnut leaf images from the village dataset	Deep convolutional neural network (DCNN)	99.88%	Springer link
[10]	2024	Arachis Hypogaea L. seed defect classification	7800 images	Combined VGG16 and InceptionV3	96.25%	Elsevier Smart Agricultural Technology
[11]	2023	Groundnut crop management by early prediction of leaf diseases	5322 real images	Deep Convolution Neural Network (CNN) model 'CNN8GN'	91.25%	International Research Journal of Multidisciplinary Technovation
[12]	2024	Automated classification of groundnut diseases using fine-tune inception V3 model	N/A	fine-tuned inception V3 model	N/A	IEEE
[13]	2024	Development of CNN Model for Detecting Groundnut Leaf Diseases Using Mobile Net	N/A	Deep learning-based model "MobileNet CNN Plant Disease Detector"	N/A	IEEE

2) EXISTING RESEARCH WORK ON PEANUT DISEASE DETECTION USING MACHINE LEARNING MODELS

Vaishnnave et al [14] used the K-Nearest Neighbor (KNN) classifier for the identification and classification of peanut leaf diseases. In their paper, researchers possessed a dataset of 250 images. They used 4 categories of diseases that are early leaf spot, late leaf spot, rust, and bud necrosis. An overall accuracy of 86% was acquired.

Guan et al [15] applied techniques such as k-nearest neighbor (KNN), and multinomial logistic regression (MLR) for the recognition of groundnut leaf spot diseases rooted on the leaf, plant, and field scale hyperspectral reflectance. Their dataset contained 1071 leaf images, 534 plant images, and 586 field images. The accuracy achieved by researchers in their research was 93.77% for leaf, 92.50% for plant, and 90.29% for field.

Ramakrishnan et al [16] explored the backpropagation algorithm method for the detection and organization the groundnut leaf diseases. In their paper, they utilized the dataset containing images of groundnut leaves affected by several diseases. They achieved an accuracy of 97.41%.

Chen et al [17] analyzed the canopy hyperspectral technique for the identification of peanut leaf spot infections. In their paper, researchers compiled the dataset from the experiment field in China. They performed two experiments to determine the Hyperspectral canopy reflectance spectra of groundnut crops susceptible to leaf spots. The study's findings suggest that adopting hyperspectral data analysis in the future will offer a reliable, efficient, and precise way to identify peanut leaf spot illnesses. The existing work for peanut diseases using machine learning has been discussed in Table 4.

TABLE 4
REVIEW OF EXISTING MODELS FOR PEANUT DISEASE IDENTIFICATION USING MACHINE LEARNING

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[14]	2019	Classification of groundnut leaf diseases using KNN classifier	250 Images	KNN classifier	86%	IEEE
[15]	2022	Detection based on Leaf, Plant, and field-scale hyperspectral reflectance	Leaf 1071 Plant 534 Field 586	Multinomial logistic regression, k-nearest neighbor	Leaf 93.77% Plant 92.50% Field 90.29%	MDPI
[16]	2015	Groundnut Leaf disease detection, and classification	N/A	Black propagation algorithm	97.41 %	IEEE
[17]	2019	Detection of peanut leaf spots disease	Experiment field, China	Canopy hyperspectral reflectance	N/A	Elsevier Computers and Electronics in Agriculture

3) EXISTING RESEARCH WORK ON PEANUT DISEASE DETECTION USING HYBRID MODELS

Seetharaman et al [18] leveraged hybrid machine-learning techniques for the real-time automatic identification of peanut diseases. Researchers compiled a collection of 129 images of peanut diseases in their paper. Six distinct classes of peanut diseases were included in their collection early leaf, Necrosis, Rust, Healthy leaf, Blight, and late spot. They attained an accuracy of about 97.3 %. Devi et al [19] explored H2K which precisely finds out and classifies groundnut leaf illnesses by using the Harris corner detector, Histogram on Oriented Gradient, and KNN classifier. Five primary groundnut illnesses were identified and categorized by researchers using a collection of 129 sample photos. They achieved 97.67% accuracy. Existing Models for Peanut Disease Identification Using a hybrid technique is discussed in Table 5.

TABLE 5
REVIEW OF EXISTING MODELS FOR PEANUT DISEASE IDENTIFICATION USING HYBRID TECHNIQUE

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[18]	2022	Real-time automatic detection	129 images	Hybrid machine learning techniques	97.3%	Springer Multimedia Tools and Application
[19]	2020	Detection and classification approach of groundnut leaf disease	129 images	H2K (Harris corner detector and HOG) and KNN	97.67%	Elsevier Computers and Electronics in Agriculture

B. PEANUT PESTS' DETECTION

In this section, we analyze existing studies on peanut pests, categorized into three main approaches: deep learning, machine learning, and hybrid models. Some sample images of pests attacking the peanut are shown in Figure 3.

FIGURE 3. Pest type detection presence on peanut crops.

1) EXISTING RESEARCH WORK ON PEANUT PEST DETECTION USING DEEP-LEARNING MODELS

P. et al [20] utilized the Groundnut Vision Transformer (GNViT) model for the image-based groundnut pest classification. They used a dataset of 4157 pest images to identify and classify pests that impact groundnut crops and reached an accuracy rate of 99.52%.

Venkatasaichandrakanthand et al [21] utilized the techniques like Enhanced Vision Transformer Architecture (EViTA) and Modified Feature Optimization (MFO) for pest identification and catgorization in peanut crops. This paper presented an Enhanced Vision Transformer Architecture (EViTA) model for pest recognition, segmentation, and classification. They leveraged the IP102 Dataset from Kaggle. They employed three pest datasets impacting peanut crops, Aphids, Wireworm, and Gram Caterpillar. They secured an accuracy rate of 98%.

Dev et al [26] used the Convolutional Neural Network (CNN) model and Enhanced Vision Transformer Architecture (EViTA) for pest detection and annotation in peanut crops. The EViTA technique improved insect picture predictions in contrast to state-of-the-art methods. Broad testing demonstration it works better than image converters, even with CNN simulations. They utilized the IP102 database of the aphids, the IP102 dataset of wireworms, and the gram moth database from Kaggle.

Sivaramakrishnan et al [23] produced a dedicated model for identifying and analyzing pest images by integrating a Convolutional Neural Network (CNN) with a VGG19 transfer learning network. Researchers leveraged an open-source dataset in their paper and scored an accuracy of 85%.

Yun et al [24] used LSCDNet (Lightweight Sandglass and Coordinate Attention Network) which is a compact deep-learning model for recognizing peanut pests. LSCDNet is a sleek model derived from DenseNet. They achieved an accuracy of 85.36% when evaluated in the practical scenarios.

Venkatasaichandrakanth et al [27] evaluated the rendition of three deep learning models namely, Convolutional Neural Network (CNN), LeNet-5, and VGG-16 for peanut pest diagnosis and categorization. In their paper, the authors used the IP102 dataset including the images of thrips, aphids, armyworms, and wireworms.

Sushma et al [25] explored several methods such as Convolutional Neural Network (CNN), Moth Flame Optimization (MFO), and Enhanced Vision Transformer Architecture (EViTA). They employed CNN to predict pest infections in peanut crops, and MFO to enhance the prediction rate by selecting the most relevant features. The dataset used in their paper consisted of 1062 total images (IP102 Dataset) out of which 565 images are of Aphids, 147 images are of Wireworm and 350 images are of Gram Caterpillar.

Shiva et al [22] used a novel Convolutional Neural Network (CNN) based model for pest detection classification in Peanut crops. The model in their paper showed exceptional results and accuracy in distinguishing 102 pest categories, drawn from a dataset of 67,714 images. They achieved a testing accuracy of 99.25%.

Deep learning based on existing research work used for the identification of the pest which attacked peanut crops has been discussed in Table 6.

TABLE 6
REVIEW OF EXISTING MODELS FOR PEANUT PEST IDENTIFICATION USING DEEP LEARNING

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[20]	2024	Groundnut pests' classification using the Vision Transformer model	4157 pest images	Groundnut Vision Transformer (GNViT) model	99.52%	PLOS ONE
[21]	2023	Pest classification and detection in peanut crops using different models	IP102 Dataset from Kaggle	Enhanced vision transformer architecture model CNN, MFO, and EViTA Algorithms	98%	IEEE
[26]	2023	Applying CNN Architecture in Peanut crops for identification and labeling of insects	IP102 Kaggle dataset	EviTA, CNN	N/A	IEEE
[23]	2024	Pest Detection using Machine Learning	Open-source dataset	CNN, VGG19 transfer learning network	85%	IEEE
[24]	2024	Research on a method for identification of peanut pests	N/A	Lightweight LSCDNet Model	85.36%	Phytopathology®
[27]	2024	Groundnut pest detection and classification using deep-learning models	IP102 dataset	CNN, LeNet-5, and VGG-16	N/A	Suranaree Journal of Science and Technology
[25]	2024	Pest Detection and Classification in Peanut Crops	1062 images from the IP102 dataset	MFO, CNN, and EviTA	N/A	International Research Journal on Advanced Engineering and Management
[22]	2024	Pest Detection and Classification in Peanut Crops	67,714 images	Using CNN, and EViTA Algorithms	99.25%	JETIR Research Journal

2) EXISTING RESEARCH WORK ON PEANUT PEST DETECTION USING MACHINE LEARNING MODELS

Tripathy et al [28] explored different mining techniques for groundnut pest thrips prediction. This experiment was done in India in which they used wireless sensory and field-level surveillance data to understand crop, weather, and pest interactions on groundnut pest Thrips. They transformed the data into important and practical insights and trained through mathematical models.

The results were used to build a prediction model and Multivariate Regression Models, which eventually culminated in the development of a forewarning system.

Divya et al [29] used regression models for groundnut pest management. In their paper, different data mining methods were utilized to transform the data into meaningful and relevant relations between the dynamic crop, weather, and pest. They created a predictive model aimed at analyzing the impact of groundnut Thrips by utilizing data mining techniques to explore the dynamic relationships between crops, weather, and pests. Table 7 describes the existing models for peanut pest identification using machine learning.

TABLE 7
REVIEW OF EXISTING MODELS FOR PEANUT PEST IDENTIFICATION USING MACHINE LEARNING

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[28]	2012	Groundnut dynamics and predictions	N/A	WSN was established, and different Data Mining techniques used	N/A	Journal of Emerging Trends in Computing and Information Sciences
[29]	2024	Peanut pest management by using data mining technique	From field	Data mining techniques, Regression models	N/A	IEEE

C. PEANUT CLASSIFICATION

In this section, we analyze existing studies on peanut classification, categorized into three main approaches: deep learning, machine learning, and hybrid models. Figure 4 shows the main four types of peanut crops.

FIGURE 4 **Sample images of four major types of peanuts.**

1) EXISTING RESEARCH WORK ON PEANUT CLASSIFICATION USING DEEP LEARNING MODELS

Yang et al [31] used a technique of "Improved VGG16" to classify the varieties of peanuts published. The model was preprocessed using a grayscale, binarization, and ROI extraction. The researchers used the data set containing 3365 with 12 classes to train the model based on the CNN Algorithm. Using the CNN model, peanut pods achieved an accuracy of 96.7% and achieved an accuracy of 90.1% on corn grain verities.

Nafiiyah et al [32] used the majority voting from the identification of Convolutional Neural Network (CNN) transfer learning models to identify the peanut type based on the leaves effectively and efficiently. The CNN transfer learning models used in this paper are Resnet101, ResNet50, ResNet18, MobileNet V2, DenseNet201, and GoogleNet. Data was collected directly from the farmer's land. The complete dataset that was used was 456 images of peanut leaves. They achieved an accuracy of 96.93%. Table 8 describes the deep learning-based research work for peanut-type classification.

TABLE 8
REVIEW OF EXISTING MODELS FOR PEANUT CLASSIFICATION USING DEEP LEARNING

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[31]	2021	Peanut variety identification and classification	3365 Images of 12 Types of Peanuts	Improved VGG16 (CNN) model	96.7%	Nature
[32]	2022	Majority voting CNN method for peanut leaf type identification	456 images of peanut leaves	CNN models	96.93%	IEEE

2) EXISTING RESEARCH WORK ON PEANUT CLASSIFICATION USING MACHINE LEARNING MODELS

Deng et al [33] aimed to train the model based on "SVM" with 2000 images that contain 20 classes. Six features were narrowed down using the technology of image processing and extracted features were 31. For classifying pictures, they have used various techniques of "Image processing", "fisher feature selection", "Support Vector Machine for classification" and "k-means" as well. They scored an accuracy rate of 92.5%. This research imposes a novel methodology that serves as a basis for subsequent applications in DUS testing and investigating peanut pedigree.

Wang et al [34] explored the technique of "Machine Vision and Neural Network". They leverage a dataset containing 1522 images to train the model using a machine learning algorithm named Artificial Neural Network (ANN). Moreover, the model acquired an accuracy of 92.5%. The outcomes illustrate that the length, length, major axis length, and perimeter exhibited the highest degree of correlation with the number of kernels.

Zhongzhi et al [35] utilized the Image Recognition technique to classify peanut pods. The researchers leveraged a dataset of 100 peanuts with one class only and each image with 50 characteristics was acquired including shape, size, color, and texture properties. The model gained an accuracy of 95% by manipulating the machine learning model of the Backpropagation Neural Network (BP-ANN).

Zhongzhi et al [36] used a dataset of 2000 pictures covering 20 published classes. In this, the experts employed Image Processing technology to distinguish the Peanut Pod's Variety. In this research, nearly 50 traits were identified for shape, color, and texture categories. The model was trained on Cluster Analysis, combined with PCA. The research concluded that there is a slight difference in categories of classification when the cumulative contribution surpassed 85%.

Zhenbo et al [37] proposed a technique of Support Vector Machine (SVM) to train the model on 263 Images that can be classified as one, two, and three peanuts. The model is trained with the accuracies of 96.72% in SVM, 81.97% in HOG and SVM, and 81.97% in the combination of Hu Invariant Moment and SVM. The paper also discusses the utilization of several machine learning techniques including cellular automata filters, Fuzzy correlation vector machine, and linear discriminant analysis method of convolution kernel function.

ZOU et al [38] used the hyperspectral imaging-based machine learning approach, with XGBoost, LightGBM, CatBoost, GBDT, and Optuna algorithms. They used a dataset containing 2000 images with 5 classes to train the model. Furthermore, the model achieved an accuracy of 99.33%. However, the Optuna algorithm was used for tuning the parameter and the optimal peanut classification model was selected through an in-depth analysis.

Sundaram et al [39] used 299 images as data and used various techniques including "Study utilizes VIS/NIR spectroscopy", "Principal Component Analysis (PCA)", and a "maximum normalization model" to classify good and bad kernels in in-shell peanuts and achieved the accuracy of 80%. The algorithm uses the reflection spectra collected from 200 pods, with each pod being shelled and evaluated for damage, discoloration, or signs of immaturity.

Narendra et al [40] developed an Intelligent System to Classify Peanuts Varieties Using K-Nearest Neighbors (K-NN) and SVM. They collected the dataset from India. In this paper, an accessible approach was proposed to assess the peanut kernel quality and identify diverse varieties of peanuts with low cost and more accuracy. The system that is proposed in this paper relies on computer vision and machine learning. They achieved an overall accuracy of 93.33% for KNN and 93.82% for the Support vector machine. A review of the existing models for peanut classification using machine learning is discussed in Table 9.

TABLE 9
REVIEW OF EXISTING MODELS FOR PEANUT CLASSIFICATION USING MACHINE LEARNING

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[33]	2018	Peanut pods variety identification and pedigree analysis	Images of 20 types of Peanuts	image processing Fisher feature selection SVM for classification K-means	92.5%	Journal of the science of food and agriculture
[34]	2014	Identification of Peanut Pods with Three or More	1522 Images	Machine Vision and Neural Network Artificial Neural Network (ANN)	92.5%	International Journal of Food Engineering
[35]	2011	Study on Origin Traceability of Peanut Pods	100 Peanuts, Huayu 22 (One Kind)	Image Recognition by BP-ANN Backpropagation Neural Network (BP-ANN)	95%	IEEE
[36]	2012	Study on pedigree clustering of peanut pod's variety	20 types of Peanuts, (2000)	Image processing cluster analysis, combined with PCA	not exactly told but 85%	IEEE
[37]	2018	Classification of peanut Images based on multi-features	263 Images, (One peanut, two peanuts, three peanuts)	Support Vector Machine (SVM)	Accuracies using different Feature Detection Methods Aspect Ratio + SVM = 96.72% HOG + SVM = 81.97% Hu Invariant Moment + SVM = 81.97%	IFAC
[38]	2022	Research on peanut variety classification	2,000 peanut samples with 400 samples from each of five varieties	hyperspectral imaging-based machine learning approach, with XGBoost, LightGBM, CatBoost, GBDT and Optuna algorithms	99.33%	SciELO - Scientific Electronic Library Online

[39]	2010	Classification of in-shell peanut kernels	Runner-type peanuts grown in Georgia during 2008, with a total of 199 in-shell peanuts in the calibration group and 100 in-shell peanuts in the validation group.	Study utilizes VIS/NIR spectroscopy, Principal Component Analysis (PCA), and a maximum normalization model to classify good and bad kernels in in-shell peanuts	80% for characterizing good and bad kernels	SpringerLink
[40]	2019	System to classify peanut varieties	From India	computer vision and machine learning (KNN and SVM)	K-nearest neighbors (93.33%) and Support vector machine (93.82%)	Springer

3) EXISTING RESEARCH WORK ON PEANUT CLASSIFICATION USING HYBRID MODELS

Balasubramaniyan et al [41] applied various deep learning techniques included techniques including Cascading Subset Feature Filtering (CSFF), Social Spider Optimization (SSO), Ada Boosting Classifier, and Adaptive Convolutional Neural Network (ACNN) for peanut classification. They used a dataset that was gathered from images of peanuts to assess color, and consistency and extract features for classification that culminated the accuracy and it reached up to 89%.

Wu et al [42] detected the peanut varieties based on hyperspectral imaging and stacked machine-learning models. In this paper, Researchers measured and contrasted the model performance of Ensemble Learning Models (SEL), extreme gradient boosting algorithm (XGBoost), light gradient boosting algorithm (LightGBM), and type boosting algorithm (CatBoost). They achieved the testing accuracy of 98.03%.

Zhu et al [43] used near-infrared spectroscopy and machine-learning algorithms to classify the peanut kernels. The methods used in this paper were Near-Infrared (NIR) Spectroscopy, Principal component analysis (PCA), linear discriminant analysis (LDA), Adaptive Boosting, and KNN. Peanut kernel samples were collected from six Chinese cities of Shandong province. They achieved an accuracy of 98.77%. Table 10 discussed the summarised form for peanut type classification using hybrid models.

TABLE 10
REVIEW OF EXISTING MODELS FOR PEANUT CLASSIFICATION USING HYBRID TECHNIQUE

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[42]	2022	Detection of peanut varieties and peanut mildew	N/A	hyperspectral imaging and stacked machine learning models	98.03%	Frontiers
[41]	2022	Color contour texture-based peanut classification for assortment identification	Data for the study is collected from peanut images to evaluate their color equivalence and extract features for classification	deep learning techniques including Cascading Subset Feature Filtering, Social Spider Optimization, Ada Boosting Classifier, and Adaptive Convolutional Neural Network for peanut classification	Up to 89%	Science Direct
[43]	2024	Origin traceability of peanut kernels	Peanut samples from six Chinese cities in Shandong province	Principal component analysis (PCA), Linear discriminant analysis (LDA), Adaptive Boosting, KNN	98.77%	Elsevier

D. PEANUT MATURITY

In this section, we analyze existing studies on peanut maturity, categorized into three main approaches: Deep learning, Machine learning, and Hybrid models. Sample images for the maturity of the peanut crops have been shown in Figure 5.

FIGURE 5 **Maturity level of Peanuts level**

1) EXISTING RESEARCH WORK ON PEANUT MATURITY USING DEEP LEARNING MODELS

Balasubramaniyan et al [44] explored various techniques to identify the maturity of peanut pods. However, it is frequently difficult to pinpoint crucial characteristics in seed quality studies, such as peanut maturity analysis. They solved this problem by examining hyperspectral sensory image features using a Cross-Layer Multi-Perception Neural Network (CLMPNN) and achieved an accuracy of 97.3%.

Williams et al [45] examined 1000 images of peanut pods with 7 classes as data and used the technique of "Convolutional Neural Networks (CNN)" to train the model. Based on color and morphological variations, a categorization system for identifying the developmental stages of fresh peanut pods is explained. Classes 1 through 7 show increasing levels of maturity with room for improvement. Existing research work for peanut maturity level has been discussed in Table 11.

TABLE 11
REVIEW OF EXISTING MODELS FOR PEANUT MATURITY USING DEEP LEARNING

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[44]	2024	Peanut maturity detection assessment based on hyperspectral sensory image feature observation	hyperspectral sensory image data of peanut pods	Cross-layer multi-perception neural network Long Short-Term Memory (LSTM) also *integrated* to improve performance	97.3% Overall Accuracy Recall Rate: 98.6% true positive rate, means how accurate identify the mature peanut pods Production Accuracy: 98.9% Reliability on model in field	Journal of Intelligent & Fuzzy Systems
[45]	1981	Method for determining peanut pod maturity	7 classes, 1000 images	Convolutional Neural Networks (CNNs)	N/A	Peanut Science

2) EXISTING RESEARCH WORK ON PEANUT MATURITY USING MACHINE LEARNING MODELS

Liang et al [46] used 315 RGB images of peanut pods obtained from the agricultural field as data and utilized the "Mahalanobis Distance Classification" technique to train the model. 315 field photos with nine distinct colour groups are used in the study to compute the pod area for the background and mesocarp. They achieved an accuracy of 94%. Reasonable estimations were used to create statistical regression for OBB and BL.

Stapleton et al [47] used a method of the "Random Forest model" for categorization and lowering the camera's wavelengths to reduce expenses and increase efficiency which examines the maturity of peanuts as well. Traditional techniques, such as removing the exocarp by hand, are time- and skill-limited.

Santos et al [48] used data from the field of 2018, and 2019 irrigated fields and the 2018 rain-fed field. The data was trained on Artificial Neural Networks (ANNs). With R2 > 0.40, the results demonstrated that linear models performed better in irrigated fields than in rainfed ones. In irrigated locations, multiple linear regressions that combined adjusted growing degree days and VIs showed higher R2 but lower RMSE for both conditions. Regardless of the ANN architecture employed, ANN models with VIs and a GDD demonstrated accuracy of R2 = 0.91 in irrigated areas. All things considered, ANN models are equally accurate at predicting peanut maturity in rainfed and irrigated regions.

Bindlish et al [49] used the "Nearest-Neighbor Classification" model for the detection of the maturity of peanut pods. The mesocarp, the middle layer of the shell, can be automatically visually analyzed to determine the maturity of peanut pods. Because the color and hardness of the mesocarp change as peanuts grow, a method that uses mesocarp color and pod size to assess pod maturity has been developed. The approach detected the number of brown-black peanuts per sample with 92.5% accuracy and obtained 66.5% precision and recall when compared to human expert classification.

Windham et al [50] used 1,379 images as data and used the technique of "Partial Least Squares Regression (PLSR)" to detect pod maturity. In order to determine the best days to dig peanuts, this study sought to create models of visible and/or visible plus shortwave near-infrared (Vis/NIR) reflectance that may predict maturity classes on a pod-by-pod basis. In 2008, 'Georgia Green peanuts were harvested for the study, and their spectra were examined using Vis/NIR reflectance spectroscopy and the hull scrape method. The findings indicated that the estimate of days before digging might be changed by three days depending on where the pods were positioned on the profile board.

Rowland et al [51] explored removing subjectivity from the pod color classification "Digital Image Model (DIM)" technique to train the model on the images with 6 classes. The DIM generates a ratio of pixel color classes by applying a color definition algorithm to a scanned image of pod mesocarp colors. The DIM exhibited greater variability when imaging pods from random places, but it was successful in predicting brown and black pods in both years. The purpose is to create an imaging system for unbiased examination and maturity prediction of peanuts. Existing work for peanut maturity using machine learning is discussed in Table 12.

TABLE 12
REVIEW OF EXISTING MODELS FOR PEANUT MATURITY USING MACHINE LEARNING

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[47]	2022	Peanut maturity classification by using feature extraction	N/A	Random forest model	N/A	IEEE
[46]	2024	Estimation of peanut maturity	consists of 315 RGB images	Mahalanobis Distance Classification Statistical Regression	Mesocarp Classification Accuracy: 94%	American Journal of Plant Sciences
[52]	2019	Peanut maturity classification using hyperspectral imagery	2016, 2017 collected peanut images	Supervised learning techniques (not given).	N/A	BioSystems Engineering
[48]	2021	Estimate Peanut Maturity Variability	2018 and 2019 irrigated fields and the 2018 rainfed field immature, medium, and fully mature categories are used	Artificial Neural Networks (ANNs) Linear Models	N/A	MDPI
[49]	2017	Assessment of Peanut Pod Maturity	N/A	Nearest-Neighbor Classification Model	66.5% Overall -> Identifying brown-black peanuts, 92.5% accuracy -> 7 maturity classes	IEEE
[50]	2010	Determination of peanut pod maturity	Total = 1,379 samples	Partial Least Squares Regression (PLSR)	N/A	ASABE
[51]	2014	Development of a digital system to evaluate Peanut Maturity	6 classes (White, Yellow 1, Yellow 2, Orange, Brown, Black)	Digital Image Model (DIM)	Range of 0.63 to 0.82	Peanut Science

3) EXISTING RESEARCH WORK ON PEANUT MATURITY USING HYBRID MODELS

Zou et al [52] aimed to train the model based on supervised learning techniques on the images collected in 2016 and 2017. The study investigates a novel hyperspectral imaging (HSI) technique for assessing peanut maturity. By using the spectrum variations between mature and immature pods, the approach does away with the requirement for subjective color assessment and laborious exocarp removal. The approach can estimate continuous-valued maturity values for individual pods and has good classification accuracy

Oliveira et al [53] proposed to determine a functional link among the peanut above ground biomass, maturity, and spectral reflectance variations over time, MTR models were constructed using Random Forest and K-nearest. They used a dataset from "Alabama USA" to train the model. Weekly samples of peanut biomass were taken, and data was gathered from two commercially irrigated fields. Two Peanut Maturity Indices were measured: brown to black and orange to black (PMI). Moreover, the model achieved accuracies of 91% and 90% for Brown-to-black pods and for orange-to-black pods.

Silvio et al [54] explored the algorithms of "Artificial Neural Network (ANN)" and "Fuzzy Inference System (FIS)" published. It sought to evaluate the model's robustness in forecasting maturity for a season not employed in the model-building process and compare findings with those of fuzzy inference systems (FIS) and artificial neural networks (ANN). The researchers utilized data collected for more than three years and accomplished an accuracy rate of 49%.
Monsef et al [55] studied the techniques of "Random Forest" and "Artificial Neural Networks (ANN)" to train the model on the images gathered in 2016 and 2017. Images of peanut plant cover at various stages of development were taken using a multispectral camera. A peanut maturity prediction model was created using the masked canopy. The accuracy of the model improved over time but declined up to 60 days after planting. For more accurate predictions, the model needs to be calibrated, perhaps by growing the database and gathering data closer to harvest. Two months following planting is when data gathering should start.
Carter et al [56] explored the dataset containing 2.3kg peanuts of 2 classes and used the technique of "Image classification" for the detection of peanut pod maturity. At the field stage, 56% of seed peanuts were in the mature brown/black color class, which is less than the ideal 70–80% range, according to a study conducted by Florida Foundation Seed Producers, Inc. (FFSP). Additionally, cleaning had a negligible effect on maturity percentages, according to the study. The maturity percentage, however, decreased to 45% in tiny pods and increased to 75% in large pods during the pre-shelling size procedure. Using the hybrid method for the maturity of peanut crops is given in Table 13.

TABLE 13
REVIEW OF EXISTING MODELS FOR PEANUT MATURITY USING THE HYBRID TECHNIQUE

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[55]	2019	Spectral canopy signature for predicting peanut maturity using multispectral imagery	2016, 2017 collected peanut images	Support Vector Machines, Random Forest, Artificial Neural Networks	N/A	Computers and Electronics in Agriculture
[53]	2024	Predicting below and above ground peanut biomass and maturity	Collected from "Alabama USA"	Random Forest (RF) and K-nearest neighbor (KNN)	Brown-to-black pods, the accuracy (100% - 9%) = 91% Orange-to-black pods, the accuracy (100% - 10%) = 90%	Computers and Electronics in Agriculture
[54]	2003	Improving peanut maturity prediction	The dataset comprises field data collected over three years (from 1992 to 1994) at the UGA Southwest Branch Station in Plains, GA.	Artificial Neural Network (ANN) Fuzzy Inference System (FIS)	49%	Lecture Notes in Computer Science
[56]	2017	Pod Maturity in the Shelling Process	2.3 kg (2 classes, Yellow and brown pods)	CNN Model	N/A	Peanut Science

IV. OVERVIEW OF BENCHMARK DATASETS

In this study, we explored datasets to support our analysis of various peanut-related conditions. Details of the existing datasets have been discussed in this section.

1) For peanut leaf diseases, we used a publicly available dataset at Mendeley Data named as "A novel groundnut leaf dataset for detection and classification of groundnut leaf diseases" which includes a total of 1,720 images across five distinct classes: *Alternaria Leaf Spot*, *Healthy*, *Leaf Spot (Early and Late)*, *Rosette*, and *Rust*. This dataset offers a balanced representation of both infected and healthy plant samples. Moreover, the data provides a strong foundation for training and evaluating models focused on disease recognition.
2) For pest detection, we utilized two datasets given below; A publicly available Kaggle IP102 dataset containing 3,422 images spread across three pest classes: *Aphids*, *Caterpillars*, and *Wireworms*. This dataset provided a solid baseline for early experimentation and benchmarking. The IP02 dataset has been used in various research before.

Domain	Dataset Name/ Type	Year	Number of Images	Reference
Peanut Diseases	A novel groundnut leaf dataset for detection and classification of groundnut leaf diseases	2024	1,720	[9]
Peanut Pest Detection	IP102	2023	3,422	[21],[26],[27],[25]

V. LIMITATIONS AND CHALLENGES

The systematic review of existing approaches in peanut agriculture has several limitations. Current peanut assessment techniques face several limitations and challenges across key areas such as maturity evaluation, types classification, disease detection, and pest management.

1) In this domain, the availability of datasets is very limited and there are needed high-quality datasets for the peanut crop yield, including disease detection, maturity analysis, pest attack detection and the classification of peanut types. The availability of correct and reliable datasets is also a challenging task that is time-consuming and demands expensive expert knowledge. Inadequate data can lead to the poor performance of the model. AI models are usually complex and can be challenging for farmers to trust. Adoption of new technology is also a challenging task for farmers.
2) Peanut classification is also challenging due to its reliance on manual sorting and lack of standardized criteria across various regions.
3) Disease detection also faces difficulties due to overlapping symptoms.
4) Pest management faces a lot of hurdles due to a lack of real-time monitoring tools. Existing models are expensive and their practical implementation is limited.
5) Many solutions are not created with input from end users (farmers/agricultural workers), resulting in an imbalance between researchers and actual agricultural activities.
6) Many studies overlook the proposed methods or crop development strategies' adverse impacts on the environment or profitability.
7) Many of the current techniques are still in the experimental or theoretical stage and are rarely used in practical farming environments.

VI. FUTURE DIRECTIONS

To address these current limitations, we will propose future directions to develop more efficient, cost-effective, and practical solutions for peanut crop management. Ensuring an accurate and comprehensive dataset collection is essential to achieve more accuracy. We'll be designing a model that will provide precise results in real-world scenarios. We'll be creating a user-friendly application especially for farmers.

This will contribute to empowering our farmers and help them diminish the anticipated problems they are facing in peanut crop management. Integrating several models provides us with better results in the future. Applications based on AI can improve crop health and provide better results. AI models have tremendous potential to improve peanut agriculture, but challenges can remain in assuring the model's robustness across different environmental conditions.

VII. CONCLUSION

In conclusion, this systematic analysis of approximately 2256 studies demonstrates the game-changing potential of smart agricultural technologies in transforming peanut cultivation approaches. Our extensive study uncovers both significant progress alongside critical challenges in the field. Although significant advancements have been made in automated systems for peanut diversity classification, maturity assessment, and disease detection, critical challenges remain, including computational complexity, limited dataset availability, and the gap between laboratory achievement and practical field implementation. Furthermore, the underground cultivation of peanut pods presents a unique challenge that demands novel technological solutions. Looking ahead, the future of peanut farming lies in crafting more integrated, user-friendly, and cost-effective solutions, with priority areas including robust dataset creation, simplified AI models that farmers can rely on, hybrid technological approaches for consistent results, and user-friendly applications that bridge the technological gap. This analysis lays the foundation for future research directions aimed at developing more sustainable and efficient peanut-growing practices, with the ultimate goal of boosting global food security while ensuring that technological breakthroughs remain accessible and practical for implementation at the farm level. Success will entirely rely on ongoing collaboration between researchers, agricultural experts, and farmers, working together to connect the theoretical progress with actual application in the field.

REFERENCES

[1] A. Sood, R. K. Sharma, and A. K. Bhardwaj, 'Artificial intelligence research in agriculture: a review', *Online Information Review*, vol. 46, no. 6, pp. 1054–1075, Sep. 2022, doi: 10.1108/OIR-10-2020-0448/FULL/PDF.

[2] Y. Zhao *et al.*, 'Whole-genome resequencing-based QTL-seq identified AhTc1 gene encoding a R2R3-MYB transcription factor controlling peanut purple testa colour', *Plant Biotechnol J*, vol. 18, no. 1, pp. 96–105, Jan. 2020, doi: 10.1111/PBI.13175.

[3] D. J. Anco *et al.*, 'Peanut yield loss in the presence of defoliation caused by late or early leaf spot', *Plant Dis*, vol. 104, no. 5, pp. 1390–1399, May 2020, doi: 10.1094/PDIS-11-19-2286-RE/ASSET/IMAGES/LARGE/PDIS-11-19-2286-RE_F5-1587161579881.JPEG.

[4] H. Liu *et al.*, 'Global transcriptome analysis of subterranean pod and seed in peanut (Arachis hypogaea L.) unravels the complexity of fruit development under dark condition', *Scientific Reports 2020 10:1*, vol. 10, no. 1, pp. 1–12, Aug. 2020, doi: 10.1038/s41598-020-69943-7.

[5] T. Dybå and T. Dingsøyr, 'Empirical studies of agile software development: A systematic review', *Inf Softw Technol*, vol. 50, no. 9–10, pp. 833–859, Aug. 2008, doi: 10.1016/J.INFSOF.2008.01.006.

[6] N. Paramanandham, S. Sundhar, and P. Priya, 'Enhancing Disease Detection With Weight Initialization and Residual Connections Using LeafNet for Groundnut Leaf Diseases', *IEEE Access*, vol. 12, pp. 91511–91526, 2024, doi: 10.1109/ACCESS.2024.3422311.

[7] S. Maheswaran, N. Indhumathi, S. Dhanalakshmi, S. Nandita, I. Mohammed Shafiq, and P. Rithka, 'Identification and Classification of Groundnut Leaf Disease Using Convolutional Neural Network', in *IFIP Advances in Information and Communication Technology*, vol. 654, Springer, Cham, 2022, pp. 251–270. doi: 10.1007/978-3-031-16364-7_19.

[8] M. P. Vaishnnave, K. Suganya Devi, and P. Ganeshkumar, ‘Automatic method for classification of groundnut diseases using deep convolutional neural network’, *Soft comput*, vol. 24, no. 21, pp. 16347–16360, Nov. 2020, doi: 10.1007/s00500-020-04946-0.
[9] B. Sasmal, A. Das, K. G. Dhal, Sk. B. Saheb, R. A. Khurma, and P. A. Castillo, ‘A novel groundnut leaf dataset for detection and classification of groundnut leaf diseases’, *Data Brief*, vol. 55, p. 110763, Aug. 2024, doi: 10.1016/j.dib.2024.110763.
[10] G. B. Gebremeskel and D. G. Mengistie, ‘Groundnut (ARACHIS HYPOGAEA L.) seed defect classification using ensemble deep learning techniques’, *Smart Agricultural Technology*, vol. 9, p. 100587, Dec. 2024, doi: 10.1016/j.atech.2024.100587.
[11] H. Kukadiya, D. Meva, Nidhi Arora, and S. Srivastava, ‘Effective Groundnut Crop Management by Early Prediction of Leaf Diseases through Convolutional Neural Networks’, *International Research Journal of Multidisciplinary Technovation*, pp. 17–31, Dec. 2023, doi: 10.54392/irjmt2412.
[12] Y. S. Sneha Sree, V. C. Reddy, and D. Raveena Judie Dolly, ‘Development of CNN Model for Detecting Groundnut Leaf Diseases Using Mobile Net’, in *2024 International Conference on Science Technology Engineering and Management (ICSTEM)*, IEEE, Apr. 2024, pp. 1–5. doi: 10.1109/ICSTEM61137.2024.10560915.
[13] A. Kaur, R. Sharma, S. Chattopadhyay, and A. Verma, ‘Automated Multiclass Classification of Groundnut Leaf Diseases Using Fine-Tuned InceptionV3 Model’, in *2024 2nd World Conference on Communication & Computing (WCONF)*, IEEE, Jul. 2024, pp. 1–4. doi: 10.1109/WCONF61366.2024.10692187.
[14] M. P. Vaishnnave, K. S. Devi, P. Srinivasan, and G. A. P. Jothi, ‘Detection and Classification of Groundnut Leaf Diseases using KNN classifier’, in *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, IEEE, Mar. 2019, pp. 1–5. doi: 10.1109/ICSCAN.2019.8878733.
[15] Q. Guan, K. Song, S. Feng, F. Yu, and T. Xu, ‘Detection of Peanut Leaf Spot Disease Based on Leaf-, Plant-, and Field-Scale Hyperspectral Reflectance’, *Remote Sens (Basel)*, vol. 14, no. 19, p. 4988, Oct. 2022, doi: 10.3390/rs14194988.
[16] Ramakrishnan M. and Sahaya Anselin Nisha A., ‘Groundnut leaf disease detection and classification by using back probagation algorithm’, in *2015 International Conference on Communications and Signal Processing (ICCSP)*, IEEE, Apr. 2015, pp. 0964–0968. doi: 10.1109/ICCSP.2015.7322641.
[17] T. Chen, J. Zhang, Y. Chen, S. Wan, and L. Zhang, ‘Detection of peanut leaf spots disease using canopy hyperspectral reflectance’, *Comput Electron Agric*, vol. 156, pp. 677–683, Jan. 2019, doi: 10.1016/j.compag.2018.12.036.
[18] Suresh and K. Seetharaman, ‘Real-time automatic detection and classification of groundnut leaf disease using hybrid machine learning techniques’, *Multimed Tools Appl*, vol. 82, no. 2, pp. 1935–1963, Jan. 2023, doi: 10.1007/s11042-022-12893-1.
[19] K. Suganya Devi, P. Srinivasan, and S. Bandhopadhyay, ‘H2K – A robust and optimum approach for detection and classification of groundnut leaf diseases’, *Comput Electron Agric*, vol. 178, p. 105749, Nov. 2020, doi: 10.1016/j.compag.2020.105749.
[20] V. P. and I. M., ‘GNViT- An enhanced image-based groundnut pest classification using Vision Transformer (ViT) model’, *PLoS One*, vol. 19, no. 3, p. e0301174, Mar. 2024, doi: 10.1371/journal.pone.0301174.
[21] P. Venkatasaichandrakanthand and M. Iyapparaja, ‘Pest Detection and Classification in Peanut Crops Using CNN, MFO, and EViTA Algorithms’, *IEEE Access*, vol. 11, pp. 54045–54057, 2023, doi: 10.1109/ACCESS.2023.3281508.
[22] B. S. M. B. T. R. K. Addanki Shiva Sai, ‘Pest Detection and Classification in Peanut Crops Using CNN, and EViTA Algorithms’, *JETIR Research Journal*, vol. 11, no. 5, May 2024.
[23] S. S, C. R. Rathish, Y. K. M, S. R, S. R. K, and E. Vanaja, ‘Pest Detection using Machine Learning’, in *2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS)*, IEEE, Apr. 2024, pp. 87–91. doi: 10.1109/ICC-ROBINS60238.2024.10533899.
[24] Y. Yun *et al.*, ‘Research on a Method for Identification of Peanut Pests and Diseases Based on a Lightweight LSCDNet Model’, *Phytopathology*, vol. 114, no. 9, pp. 2162–2175, Sep. 2024, doi: 10.1094/PHYTO-01-24-0013-R.
[25] Sushma D S, Mohammed Alqhama, Aravind M, Jayanth A B, and Rakshith Kumar K, ‘Pest Detection and Classification in Peanut Crops’, *International Research Journal on Advanced Engineering and Management (IRJAEM)*, vol. 2, no. 05, pp. 1372–1379, May 2024, doi: 10.47392/IRJAEM.2024.0189.
[26] S. Dev, ‘Application of CNN Architecture for Insect Identification and Labeling in Peanut Crops’, in *2023 International Conference on Data Science and Network Security (ICDSNS)*, IEEE, Jul. 2023, pp. 1–8. doi: 10.1109/ICDSNS58469.2023.10244877.
[27] V. P and I. M., ‘GROUNDNUT CROP PEST DETECTION AND CLASSIFICATION USING COMPREHENSIVE DEEP-LEARNING MODELS’, *Suranaree Journal of Science and Technology*, vol. 31, no. 1, pp. 020028(1–17), Feb. 2024, doi: 10.55766/sujst-2024-01-e02544.
[28] J. A. S. N. M. U. B. D. K. V. D. R. S. N. M. H. T. K. A. K. Tripathy, ‘Data Mining and Wireless Sensor Network for Groundnut Pest Thrips Dynamics and Predictions ’, *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, Jun. 2012.
[29] M. Divya, T. N. Manjunath, and R. S. Hegadi, ‘A Study on Developing Analytical Model for Groundnut Pest Management Using Data Mining Techniques’, in *2014 International Conference on Computational Intelligence and Communication Networks*, IEEE, Nov. 2014, pp. 691–696. doi: 10.1109/CICN.2014.152.
[30] A. and D. S. and V. K. and M. S. and D. U. {Jagarlapudi, ‘Data Mining and Wireless Sensor Network for Groundnut Pest/Disease Interaction and Predictions’, *International Journal of Computer Information Systems and Industrial Management Applications*, pp. 427–436, Dec. 2013.
[31] H. Yang, J. Ni, J. Gao, Z. Han, and T. Luan, ‘A novel method for peanut variety identification and classification by Improved VGG16’, *Sci Rep*, vol. 11, no. 1, p. 15756, Aug. 2021, doi: 10.1038/s41598-021-95240-y.
[32] N. Nafiiyah, N. S. Fatonah, R. Wardhani, B. Jokonowo, and T. A. Y. Siswa, ‘Majority voting transfer learning CNN for peanut leaf types identification’, in *2022 3rd International Conference on Electrical Engineering and Informatics (ICon EEI)*, IEEE, Oct. 2022, pp. 70–74. doi: 10.1109/IConEEI55709.2022.9972271.
[33] L. Deng and Z. Han, ‘Image features and DUS testing traits for peanut pod variety identification and pedigree analysis’, *J Sci Food Agric*, vol. 99, no. 5, pp. 2572–2578, Mar. 2019, doi: 10.1002/jsfa.9472.
[34] J. Wang, W. W. Yang, L. T. Walker, and T. Rababah, ‘Identification of Peanut Pods with Three or More Kernels by Machine Vision and Neural Network’, *International Journal of Food Engineering*, vol. 10, no. 1, pp. 97–102, Jan. 2014, doi: 10.1515/ijfe-2012-0009.
[35] Han Zhongzhi, Deng Limiao, and Yu Renshi, ‘Study on origin traceability of peanut pods based on image recognition’, in *2011 International Conference on System science, Engineering design and Manufacturing informatization*, IEEE, Oct. 2011, pp. 93–96. doi: 10.1109/ICSSEM.2011.6081338.
[36] H. Zhongzhi, Z. Qin, L. Jing, and Y. Renshi, ‘Study on the Pedigree Clustering of Peanut Pod’s Variety Based on Image Processing’, in *2012 Fifth International Conference on Information and Computing Science*, IEEE, Jul. 2012, pp. 41–44. doi: 10.1109/ICIC.2012.54.
[37] Z. Li, B. Niu, F. Peng, G. Li, Z. Yang, and J. Wu, ‘Classification of Peanut Images Based on Multi-features and SVM’, *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 726–731, 2018, doi: 10.1016/j.ifacol.2018.08.110.
[38] Z. ZOU, L. WANG, J. CHEN, T. LONG, Q. WU, and M. ZHOU, ‘Research on peanut variety classification based on hyperspectral image’, *Food Science and Technology*, vol. 42, 2022, doi: 10.1590/fst.18522.

[39] J. Sundaram, C. V. K. Kandala, and C. L. Butts, ‘Classification of in-shell peanut kernels nondestructively using VIS/NIR reflectance spectroscopy’, *Sens Instrum Food Qual Saf*, vol. 4, no. 2, pp. 82–94, Jun. 2010, doi: 10.1007/s11694-010-9098-9.

[40] V. G. Narendra and K. Govardhan Hegde, ‘Intelligent System to Classify Peanuts Varieties Using K-Nearest Neighbors (K-NN) and Support Vector Machine (SVM)’, Springer, 2019, pp. 359–368. doi: 10.1007/978-981-15-0108-1_33.

[41] M. Balasubramaniyan and C. Navaneethan, ‘Color contour texture based peanut classification using deep spread spectral features classification model for assortment identification’, *Sustainable Energy Technologies and Assessments*, vol. 54, p. 102524, Dec. 2022, doi: 10.1016/j.seta.2022.102524.

[42] Q. Wu *et al.*, ‘Rapid nondestructive detection of peanut varieties and peanut mildew based on hyperspectral imaging and stacked machine learning models’, *Front Plant Sci*, vol. 13, Nov. 2022, doi: 10.3389/fpls.2022.1047479.

[43] R. Zhu, X. Wu, B. Wu, and J. Gao, ‘High-accuracy classification and origin traceability of peanut kernels based on near-infrared (NIR) spectroscopy using Adaboost - Maximum uncertainty linear discriminant analysis’, *Curr Res Food Sci*, vol. 8, p. 100766, 2024, doi: 10.1016/j.crfs.2024.100766.

[44] M. Balasubramaniyan and C. Navaneethan, ‘Peanut maturity detection assessment using cross-layer multi-perception neural network based on hyperspectral sensory image feature observation’, *Journal of Intelligent & Fuzzy Systems*, vol. 46, no. 4, pp. 9393–9407, Apr. 2024, doi: 10.3233/JIFS-239332.

[45] E. J. Williams and J. S. Drexler, ‘A Non-Destructive Method for Determining Peanut Pod Maturity’, *Peanut Science*, vol. 8, no. 2, pp. 134–141, Jul. 1981, doi: 10.3146/i0095-3679-8-2-15.

[46] W.-Z. Liang, K. R. Kirk, J. S. Thomas, and A. C. Warner, ‘Estimation of Peanut Maturity Using Color Image Analysis’, *Am J Plant Sci*, vol. 15, no. 08, pp. 617–635, 2024, doi: 10.4236/ajps.2024.158042.

[47] S. N. B. Tushar, S. Sarker, W. Stapleton, and D. Valles, ‘Peanut maturity classification by features extracted from selected hyperspectral components’, in *2022 IEEE Global Humanitarian Technology Conference (GHTC)*, IEEE, Sep. 2022, pp. 176–183. doi: 10.1109/GHTC55712.2022.9911049.

[48] A. F. Santos *et al.*, ‘Using UAV and Multispectral Images to Estimate Peanut Maturity Variability on Irrigated and Rainfed Fields Applying Linear Models and Artificial Neural Networks’, *Remote Sens (Basel)*, vol. 14, no. 1, p. 93, Dec. 2021, doi: 10.3390/rs14010093.

[49] E. Bindlish, A. L. Abbott, and M. Balota, ‘Assessment of Peanut Pod Maturity’, in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Mar. 2017, pp. 688–696. doi: 10.1109/WACV.2017.82.

[50] W. R. Windham, C. V. K. Kandala, J. Sundaram, and R. C. Nuti, ‘Determination of Peanut Pod Maturity by Near-Infrared Reflectance Spectroscopy’, *Trans ASABE*, vol. 53, no. 2, pp. 491–495, 2010, doi: 10.13031/2013.29558.

[51] B. C. Colvin, D. L. Rowland, J. A. Ferrell, and W. H. Faircloth, ‘Development of a Digital Analysis System to Evaluate Peanut Maturity’, *Peanut Science*, vol. 41, no. 1, pp. 8–16, Jan. 2014, doi: 10.3146/PS13-9.1.

[52] S. Zou, Y.-C. Tseng, A. Zare, D. L. Rowland, B. L. Tillman, and S.-C. Yoon, ‘Peanut maturity classification using hyperspectral imagery’, *Biosyst Eng*, vol. 188, pp. 165–177, Dec. 2019, doi: 10.1016/j.biosystemseng.2019.10.019.

[53] M. F. Oliveira *et al.*, ‘Predicting below and above-ground peanut biomass and maturity using multi-target regression’, *Comput Electron Agric*, vol. 218, p. 108647, Mar. 2024, doi: 10.1016/j.compag.2024.108647.

[54] H. L. Silvio, R. W. McClendon, and E. W. Tollner, ‘Improving Peanut Maturity Prediction Using a Hybrid Artificial Neural Network and Fuzzy Inference System’, 2000, pp. 543–548. doi: 10.1007/3-540-45049-1_64.

[55] H. Abd-El Monsef, S. E. Smith, D. L. Rowland, and N. Abd El Rasol, ‘Using multispectral imagery to extract a pure spectral canopy signature for predicting peanut maturity’, *Comput Electron Agric*, vol. 162, pp. 561–572, Jul. 2019, doi: 10.1016/j.compag.2019.04.028.

[56] E. T. Carter *et al.*, ‘Pod Maturity in the Shelling Process’, *Peanut Science*, vol. 44, no. 1, pp. 26–34, Jan. 2017, doi: 10.3146/PS16-17.1.

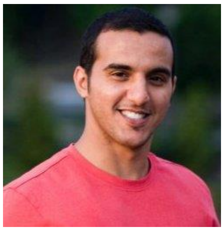
UME HABIBA is a Software Engineering student at the University of Sargodha, Pakistan, motivated by an affinity for solving complex problems through technological innovation. She was born in Sargodha, Pakistan, in 2002. She completed her intermediate education at ILM Group of Colleges, Sargodha Campus, and has actively contributed to projects advancing intelligent systems. Her research interests span Big Data Analytics, Machine Learning, Deep Learning, and Natural Language Processing (NLP), applied to domains such as Smart Agriculture and Healthcare, with a specialized focus on AI-driven cancer research. Committed to interdisciplinary impact, she aims to bridge emerging technologies with societally critical challenges, leveraging tools like AI and data science to drive scalable solutions.

QAISAR ABBAS, is contributing in the field of academics and research as a faculty member. His area of interest includes Natural Language Processing / Computational Linguistics, Machine/Deep Learning, Mathematical & Statistical Modelling, and Information Retrieval in the domain of Artificial Intelligence. He obtained his MS in Computer Science in 2010 from National University of Computer & Emerging Sciences, Lahore, Pakistan and then he completed his doctoral studies in Natural Language Processing / Computational Linguistics in 2014 from University of Konstanz, Germany. Currently, he is serving in Islamic University, Madinah, Saudi Arabia. He is an active member of many research-oriented societies & collaborations and contributing his work in quality conferences and journals. His profile can be accessed from the following URL: www.clsp.org/qabbas

NABEEL ABBAS was born in Chakwal, Pakistan in 2003. He is currently a student of Software Engineering at the University of Sargodha, Pakistan. He did his intermediate education in Pre-Engineering from the Punjab Group of Colleges, Chakwal Campus. His research interests include Big Data Analytics, Artificial Intelligence, Computer Vision and Smart Agriculture with a focus on producing highly efficient and scalable AI solutions. He is primarily interested in the creation of novel AI-driven models to drive sustainable agricultural solutions. Through multiple research projects, he utilizes advanced computational approaches to tackle challenging problems in the sector of agriculture and beyond. Nabeel has a tremendous interest in contributing to real-world problems by utilizing the powerful tools of AI and Data Science to create innovative solutions.

TURKI ALGHAMDI received the B.Sc. degree in computer science, in 2005, the M.Sc. degree in software engineering, in 2008, and the Ph.D. degree in software engineering from De Montfort University, U.K., in 2012. He is currently working as an Associate Professor with the Faculty of Computer and Information Systems, Islamic University of Madinah, Saudi Arabia. His research interests include big data analysis, the IoT, wireless sensor networks, and IS policy management.

KASHIF MAHMOOD was born in the historic city of Chiniot, Pakistan in 2003. He accomplished his intermediate education in Computer Science at the Superior Group of colleges, Chiniot Campus. He is currently pursuing a degree in Software Engineering from the University of Sargodha, Pakistan. His research interests encompass Artificial Intelligence, especially in areas like Machine Learning, Deep Learning, Language Processing, Computer Vision, and Data Science, with applications spanning Healthcare, Smart Agriculture, and beyond. He is particularly focused on developing advanced AI-driven models and algorithms that enhance efficiency, scalability and accuracy across diverse domains. Committed to real-world impact, Kashif is passionate about integrating emerging technologies with real-world challenges, leveraging AI and Data Science to drive scalable and transformative solutions.



SAMI S. ALBOUQ received the Bachelor's, Master's, and Doctoral degrees in Computer Science in 2009, 2011, and 2018, respectively. He currently serves as an Associate Professor at the College of Computer Science at the Islamic University of Madinah, Al-Madinah, Saudi Arabia. His research interests include system security engineering, complex information technology systems, and quantum key distribution systems.

MUHAMMAD RAMZAN has completed the PhD (Computer Science) degree at the University of Management and Technology, Lahore, Pakistan. He is currently a Lecturer at the University of Sargodha, Pakistan. He has authored more than 50 research articles published in reputed peer-reviewed journals. His areas of research include Artificial Intelligence, Machine Learning, Deep Learning, Digital Image Processing, Medical imaging, Video surveillance, Human Activity Recognition, and computer vision.

Dr SAAD ALQAHTANY, Assistant Professor in Cyber Security & digital forensics at Faculty of Computer Science & Information systems in Islamic University of Al-Madinah. Alqahtany achieved his PhD in Cyber security and Digital Forensics from Plymouth University (UK) in 2017. Saad achieved a master's degree in digital Forensics from Bradford University (UK) in 2010. He graduated from King Saud University (Riyadh) in 2002 and achieved his BSc in Computer Science. Saad has been working for Islamic University as an Assistant Professor in Cyber Security & digital forensics since 2021. He was working for the Interior ministry of Saudi Arabia for over 19 years. He has been holding many positions related to cyber security and digital investigation. His professional interests focus on Digital Forensics, Cyber Security, Cybercrimes Investigations, IoT Forensics, and Cloud Forensics. Current interest is to enable a cyber infrastructure for a digital investigation and fighting cyber criminals in a forensically and proactively manner. Experienced Head of Cyber Crime and Digital Forensics Unit with a demonstrated history of working in the computer and network security industry. Saad has published several papers and journals in local and international conferences.

sensors

MDPI

Type of the Paper (Article, Review, Communication, etc.)

Utilizing Pre-Trained Models for Peanut Pest Identification through Deep Learning Techniques

Nabeel Abbas [1], Kashif Mahmood [2] and Muhammad Ramzan [2,*]

[1] Affiliation 1; e-mail@e-mail.com
[2] Affiliation 2; e-mail@e-mail.com
* Correspondence: e-mail@e-mail.com; Tel.: (optional; include country code; if there are multiple corresponding authors, add author initials)

Abstract: Agriculture plays an important role in enhancing the economy of any country. It provides food security and contributes to national income. Farming technology is being used to solve complex problems, such as pest attack identification. Peanuts have importance among various crops as they directly impact the economy of a country by providing food security and oil seeds. They are often damaged due to different kinds of pests. These pests cause major losses for farmers as they can negatively affect the quality and quantity of peanuts. So, it is very important to detect the pests in the early stage to enhance production and save them from major losses. In this paper, we simply propose an efficient method to detect pests using image processing and deep learning models. We used a publicly available dataset of peanuts. It has different kinds of images, such as healthy and infected with different pests. These images were captured in different lightning and background conditions. We marked the images based on pest type and symptoms. We trained a deep learning model with the help of this dataset to identify patterns and infected areas in the peanut crop images. Then, we tested our model to assess the accuracy. The results show high accuracy for detecting the kinds of pests. Our proposed system can assist the farmers' community in examining the peanut crops more effectively. It also saves time from manual monitoring and provides more accurate decisions. This research focuses on smart agriculture by developing an effective solution. In the future, our system can also be extended to mobile applications or drone-based monitoring systems. This paper promotes sustainable agriculture and solves real-world challenges related to agriculture.

Keywords: Pest Identification; Deep Learning Models; Pre-Trained Models; Artificial Intelligence; Image Processing

Academic Editor: Firstname Lastname

Received: date
Revised: date
Accepted: date
Published: date

Citation: To be added by editorial staff during production.

1. Introduction

Agriculture is considered a fundamental pillar of any country as it can be a way to enhance social and economic development. The demand for food is increasing day by day, and it becomes challenging to increase the production of crops due to several reasons. Peanut is a significant oil crop worldwide, fulfilling the demands of people in various ways. But there are different threats to this crop, and pest infestation remains an important one. Pests decrease the quantity as well as the quality. It leads to major losses, such as economic and food supply. Peanut crops are particularly highly valued due to their nutritional significance. But they are frequently attacked by pests such as aphids, wireworms, and caterpillars, which can cause visible damage to fruits and leaves [1].

Pest detection in peanut crops was performed by farmers manually in the past. However, this technique is often time-consuming. It also requires a large workforce and causes human errors. [2]. The increasing size of the field makes it difficult to inspect every plant individually, which can result in late identification of insects. If it is not resolved in the early stage, then it can lead to heavier crop damage and also cause the use of pesticides. The chemicals used in pesticides may harm the environment and human health badly. [3].

To address these issues, researchers have developed many solutions, including the use of Artificial intelligence. Modernization in AI, especially in image processing and deep learning techniques, makes it possible to detect pests automatically. [4]. These methods can provide more accurate and cost-effective results in the identification of pests from images. Moreover, it reduces the demand for a large number of laborers and provides real-time monitoring of large areas. [5].

We propose a deep learning model to detect pests in peanut crops using image processing techniques. We use a publicly available dataset of IP02 containing healthy and pest-infected images. These images were captured under different lighting conditions and backgrounds to reflect real-world scenarios. Furthermore, we train a deep learning model using this dataset to find types of pests and also the affected regions in the crop. The major contributions of this paper are:

- Development of a deep learning model for pest detection using a publicly available dataset, along with the creation of a user-friendly application that allows users to detect pest types through images
- The creation of a user-friendly application that allows users to detect pest types using image classification techniques.
- Enhancing the accuracy by training and evaluation of the model.
- Elevating the existing data.
- Provides the discussions for farmers about how this approach can support them in identifying pests in the early stage. Our model supports the friendly-environment practices to diminish the overuse of chemical pesticides.

The remaining paper is organized as follows: Section 2 reviews related work in pest detection and deep learning applications in agriculture. Section 3 explains the dataset and methodology used. Section 4 shows the experimental results and performance evaluation. Finally, Section 6 concludes the paper.

2. Literature Review

It is very crucial to detect pests effectively to ensure peanut crop health. This section categorizes existing research into three major approaches: machine learning (ML)-based methods, deep learning (DL)-based methods, and hybrid models.

2.1. Existing papers on pest identification using deep learning models

Venkatasaichandrakanth et al [6] proposed the GNViT model, built on a pre-trained Vision Transformer (ViT) architecture, with data augmentation and fine-tuning for groundnut pest classification. They used the dataset of IP02 from Kagggle and achieved 99.52% accuracy. Thakare et al [7] proposed an advanced pest detection strategy that employs a hybrid optimization-tuned deep convolutional neural network (CNN) for accurate identification of pests in agricultural fields, utilizing wireless sensor networks and optimized feature extraction methods to achieve high detection accuracy. They identified pests from the habitus image database, and they achieved an accuracy of 94.35%. Dev et al [8] used the Convolutional Neural Network (CNN) model and Enhanced Vision Transformer Architecture (EViTA) for pest detection and annotation in peanut crops. The

EViTA technique improved insect picture predictions in contrast to state-of-the-art methods. Broad testing demonstrates it works better than image converters, even with CNN simulations. They utilized the IP102 database of the aphids, the IP102 dataset of wireworms, and the gram moth database from Kaggle. Sivaramakrishnan et al [9]produced a dedicated model for identifying and analyzing pest images by integrating a Convolutional Neural Network (CNN) with a VGG19 transfer learning network. Researchers leveraged an open-source dataset in their paper and scored an accuracy of 85%. Yun et al [10] used LSCDNet (Lightweight Sandglass and Coordinate Attention Network), a compact deep-learning model for recognizing peanut pests. LSCDNet is a sleek model derived from DenseNet. They achieved an accuracy of 85.36% when evaluated in the practical scenarios. Venkatasaichandrakanth et al [11] assessed the rendition of three deep learning models, namely, Convolutional Neural Network (CNN), LeNet-5, and VGG-16, for peanut pest diagnosis and categorization. In their paper, the authors used the IP102 dataset, including the images of thrips, aphids, armyworms, and wireworms. Sushma et al [12]explored several methods such as Convolutional Neural Network (CNN), Moth Flame Optimization (MFO), and Enhanced Vision Transformer Architecture (EViTA). They employed CNN to predict pest infections in peanut crops, and MFO to enhance the prediction rate by selecting the most relevant features. The dataset used in their paper consisted of 1062 total images (IP102 Dataset), out of which 565 images are of Aphids, 147 images are of Wireworm, and 350 images are of Gram Caterpillar.

Deep learning, based on existing research work used for the identification of the pest that attacks peanut crops has been discussed in Table 1.

Table 1. Existing papers on peanut pests based on deep learning techniques.

Ref	Article Name	Dataset	Method	Accuracy	Journal Name
[6]	GNViT – An Enhanced Image-Based Groundnut Pest Classification Using Vision Transformer (ViT) Model (2024)	IP102 Dataset from Kaggle	Groundnut Vision Transformer (GNViT) model	99.52%	PLOS ONE
[7]	Advanced Pest Detection Strategy Using Hybrid Optimization Tuned Deep Convolutional Neural Network (2022)	Habitus Image Database	Deep CNN model	94.35%	Journal of Engineering, Design and Technology
[8]	Applying CNN Architecture in Peanut crops for identification and labeling of insects (2023)	IP102 Kaggle dataset	EviTA, CNN	N/A	IEEE
[9]	Pest Detection using Machine Learning (2024)	Open-source dataset	CNN, VGG19 transfer learning network	85%	IEEE
[10]	Research on a method for the identification of peanut pests (2024)	N/A	Lightweight LSCDNet Model	85.36%	Phytopathology®
[11]	Groundnut pest detection and classification using deep-learning models (2024)	IP102 dataset	CNN, LeNet-5, and VGG-16	N/A	Suranaree Journal of Science and Technology
[12]	Pest Detection and Classification in Peanut Crops (2024)	1062 images from the IP102 dataset	MFO, CNN, and EviTA	N/A	International Research Journal on Advanced Engineering and Management

2.2. Existing papers on pest identification using machine learning models

Tripathy et al [13] explored different mining techniques for groundnut pest thrips prediction. This experiment was done in India, in which they used wireless sensory and field-level surveillance data to understand crop, weather, and pest interactions on groundnut pest Thrips. They transformed the data into important and practical insights and trained through mathematical models. The results were used to build a prediction model and Multivariate Regression Models, which eventually culminated in the development of a forewarning system. Divya et al [14] used regression models for groundnut pest management. In their paper, different data mining methods were utilized to transform the data into meaningful and relevant relations between the dynamic crop, weather, and pest. They created a predictive model aimed at analyzing the impact of groundnut Thrips by utilizing data mining techniques to explore the dynamic relationships between crops, weather, and pests. Table 2 describes the existing models for peanut pest identification using machine learning. Existing papers on peanut pest identification based on machine learning models are shown in Table 2.

Table 2. Existing papers on peanut pests based on machine learning techniques.

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[13]	2012	Groundnut dynamics and predictions	N/A	WSN was established, and different Data Mining techniques were used	N/A	Journal of Emerging Trends in Computing and Information Sciences
[14]	2014	Peanut pest management by using a data mining technique	From field	Data mining techniques, Regression models	N/A	IEEE

2.3. Existing papers on pest identification using hybrid models

Venkatasaichandrakanth et al [15] Provided a comprehensive review of artificial intelligence techniques for pest detection and classification, discussing both machine learning and deep learning approaches. They emphasized the advantages of using hybrid models, including deep learning methods like convolutional neural networks (CNNs), in improving the accuracy of pest detection in crops. The study utilized the Microsoft Academic Graph (MAG) and created an evaluation dataset comprising 1,000 sentences annotated with method and dataset entities. The authors developed a framework that includes domain-specific Named Entity Recognition (NER) to identify method and dataset mentions, followed by classification into 'used' vs. 'non-used' categories. They experimented with various models, including Random Forests with TF-IDF features, Random Forests with SciBERT embeddings, fine-tuned SciBERT, and SciBERT combined with Convolutional Neural Networks. The fine-tuned SciBERT model achieved an F1-score of 0.82 for method usage classification and 0.81 for dataset usage classification. The past work related to pests using hybrid approaches is labeled in Table 3.

Table 3. Existing papers on peanut pests based on hybrid techniques.

Ref	Year	Article Name	Dataset	Method	Accuracy	Journal Name
[15]	2021	Identifying Used Methods and Datasets in Scientific Publications	1,000 annotated sentences from Microsoft Academic Graph (MAG)	NER-based framework with Random Forest, SciBERT, and SciBERT-CNN models	SciBERT achieved F1-scores of 0.82 (method) and 0.81 (dataset) for usage classification.	CEUR Workshop Proceedings

3. Proposed Research Methodology

The proposed research methodology is an important section as it provides the complete details of the system. It outlines the procedures, tools, and methods to gather and process data. The complete description of the models is also mentioned. This section provides the details of each component, including data acquisition, pre-processing, feature extraction, classification, and performance measurement. Figure 1 represents the overall methodology of our proposed models used in our research work.

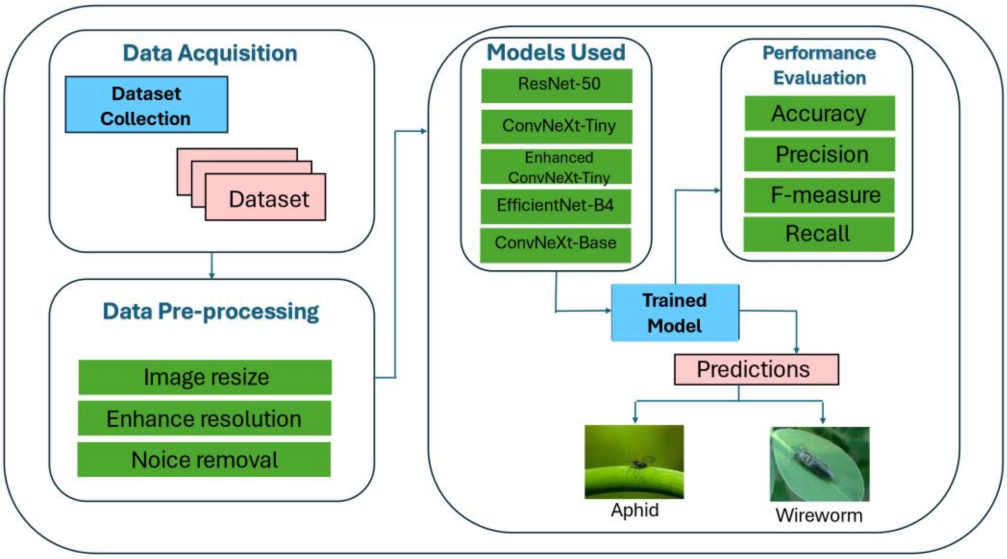


Figure 1. Proposed architecture workflow of Peanut disease detection

3.1. Data Acquisition for our work

A publicly available dataset was used for the task of peanut pest detection to ensure reproducibility and facilitate comparative evaluation. Specifically, the IP102 dataset from Kaggle was utilized, which is well-known in the field of agricultural pest classification. Figure 2 includes the three classes of pest images present in our dataset.

(a) (b) (c)

Figure 2. Three classes of pests included in our dataset: (a) Caterpillar, (b) Wireworm, (c) Aphids

The dataset contains a total of 3,422 high-quality images, distributed across three key pest classes commonly affecting peanut crops. These classes are Caterpillars, Wireworms, and Aphids. Each image in the dataset is labelled and categorized, enabling supervised training of classification models. The IP102 dataset has been extensively used in previous research, making it a reliable benchmark for performance validation and model comparison. Its structured format and diverse samples offer a solid foundation for developing and testing deep learning models for pest identification in precision agriculture. Moreover, the name of each folder is the same as the corresponding image class. The Caterpillars class contains all the images related to this pest. Photos of wireworms affecting peanuts are given in the folder named Wireworms. In the folder Aphids, all pictures of aphids are uploaded. Additionally, we created a custom dataset by adding two more classes to the previous IP102 dataset. These two classes are Armyworm and Thrips. Hence total of 5 classes are utilized in this dataset. The dataset comprises a total of 4591 images. Images were divided as 80% training, 10% testing, and 10% validation accuracy images in both datasets. Figure 3 includes the five classes of pest images present in our dataset.

(a) (b) (c) (d) (e)

Figure 3. Five classes of pests (self-created) included in our dataset: (a) Caterpillar, (b) Wireworm, (c) Aphids, (d) Armyworm, (e) Thrips

Table 4. Details of the dataset in details being used in our work.

Class name	Number of images	Images in the train folder	Images in the test folder	Images in the value accuracy folder
Caterpillar	434	347	44	43
Wireworm	532	426	53	53
Aphids	2456	1965	245	246

Armyworm	642	514	64	64
Thrips	527	422	53	52
Total	4591	3674	459	458

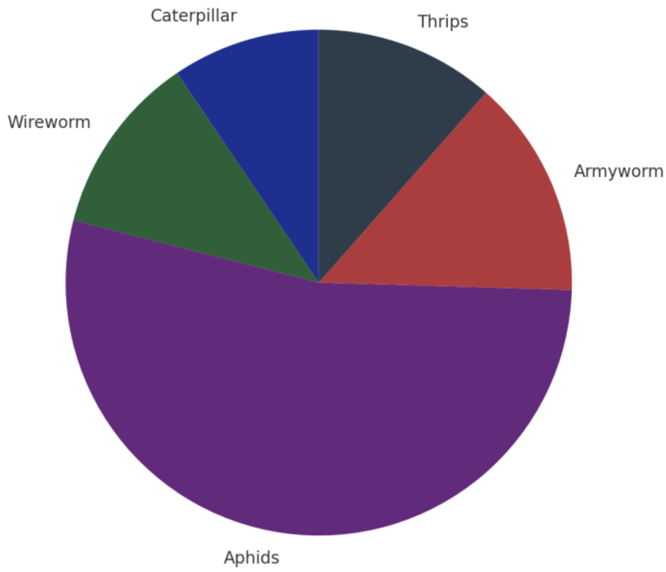


Figure 4. Overview of the dataset using a pie chart

3.2. Pre-processing of images

It is very crucial to pre-process the data as it enhances the effectiveness of deep learning models. The images in the IP102 dataset were passed through a series of preprocessing techniques to improve model performance and reduce noise and irrelevant variations in the data. First of all, the images were resized to a resolution of 224 × 224 pixels to ensure compatibility with the deep learning model. To train the model more efficiently, the pixel intensity values were set to a [0, 1] scale by dividing each pixel value by 255. A Gaussian blur was applied to reduce image noise. Different data augmentation techniques, such as flipping, rotations, zooming, and brightness adjustments, were applied due to the limited dataset size. Pest class labels, such as Aphids, Caterpillars, and Wireworms, were converted into a numerical format due to multiclass classification. These preprocessing steps ensured that the dataset was clean, consistent, and suitably structured for input into the feature extraction and classification pipeline.

3.3. Feature extraction and classification

In our research work, we utilized four deep learning pre-trained models, such as ConvNeXt-Base, ConvNeXt-Tiny, Enhanced ConvNeXt-Tiny, EfficientNet-B4, and ResNet-50.

3.3.1. ConvNeXt-Base Model Architecture

ConvNeXt is a convolutional neural network architecture proposed by Zhuang Liu et al. in the paper *"A ConvNet for the 2020s,"* [16] published in 2022. The model rethinks and modernizes classic ConvNet designs by integrating architectural insights from vision transformers (ViTs), resulting in a family of models (ConvNeXt-Tiny to ConvNeXt-XL) that match or surpass the performance of transformer-based models on various vision benchmarks. ConvNeXt-Base is one of the intermediate variants offering a balanced trade-off between performance and efficiency. ConvNeXt-Base adopts a pure convolutional design but incorporates many ViT-inspired enhancements, such as Patchify Stem, Depthwise Separable Convolutions, Large Kernel Sizes, Layer Normalization (LN), GELU Activation, and Stochastic Depth and Layer Scaling. The table below presents the stage-wise configuration of this model. Each ConvNeXt block includes depthwise convolution, LayerNorm, a pointwise convolution (1×1), GELU activation, and a residual connection. Table 5, given below, explains the detailed stage-wise architecture.

Table 5. Stage-wise configuration of ConvNeXt-Base Model

Stage	Resolution	Operator	# Layers	#Channels
Stem	224×224	Conv (4×4, s=4)	1	96
1	56×56	ConvNeXt Block	3	96
2	28×28	ConvNeXt Block	3	192
3	14×14	ConvNeXt Block	9	384
4	7×7	ConvNeXt Block	3	768
Head	7×7	Global Avg Pool + FC	1	1000

The model employs a 4-stage hierarchical structure with progressive downsampling through strided convolutions:

$$Stage_i\ (x) = \ ConvNeXtBlock^{N_i}\ \ (Patchify(x)) \tag{1}$$

where N1=3N1=3, N2=3N2=3, N3=27N3=27, N4=3N4=3 blocks per stage, and patchify layers use 4×4 kernels for spatial reduction.
Each ConvNeXt Block implements:

$$x_{out} = x + DW(Conv(GELU(LayerNorm(MLP(x))))) \tag{2}$$

utilizing 7×7 depthwise convolutions for large receptive fields and inverted bottlenecks with 4× expansion ratios. The base configuration contains 88M parameters, achieving 84.1% ImageNet accuracy through its macro design that mimics Swin Transformers' stage compute distribution.

3.3.2. EfficientNet-B4 Model Architecture

EfficientNet-B4 was first proposed by Mingxing Tan and Quoc V. Le in the paper "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks" [17] Prior to EfficientNet, known scaling of CNNs mostly came via increasing depth, width, or resolution individually. However, this process had no systematic methodology. To address this issue, Google Brain researchers introduced the EfficientNet model family based upon a compound scaling method. It scales in all three dimensions proportionally using defined scaling coefficients to get optimal performance without separately increasing depth, width or resolution. The EfficientNet family has medium sized model named as EfficientNet B4 which is an adjustment in between the accuracy and computation efficacy. It was trained using a multi-objective neural architecture search with the same aim as MnasNet, of optimizing accuracy as well as FLOPs. Specifically, the architecture is as follows:

- **Stem Layer:** 3×3 convolution with stride of 2, padding of 1 on input image to get 48 channels which is output from Stem Layer. It is this layer, which is the first to take care of the initial feature extraction.
- **IRB Layer:** The total number of IRB blocks is 14, each containing a (depthwise separable) convolution layer, an expansion layer, a (depthwise) 3 × 3 convolution layer, and a squeeze-and-excitation (SE) block. Across blocks, number of channels, expansion ratio and stride are different. For instance, the first IRB block has 24 channels, an expansion ratio of 1, a stride of 1 and has an SE module.
- **Head Layer:** output of these blocks is then fed to 1×1 convolution of 1792 channels and 7×7 adaptive average layer of pooling to remove the spatial dimension of maps feature.

- **Fully Connected Layer:** This means that the output of the Head layer is flattened and passed to a Fully Connected layer with 1000 neurons for a final classification since we will be using 1000 neurons as per the task.

Table 6. Stage-wise configuration of EfficientNet-B4 Model

Stage	Operator	Resolution	#Channels	#Layers
1	Conv3x3	380 × 380	48	1
2	MBConv1, k3x3	190 × 190	24	1
3	MBConv6, k3x3	190 × 190	32	2
4	MBConv6, k5x5	95 × 95	56	2
5	MBConv6, k3x3	48 × 48	112	3
6	MBConv6, k5x5	24 × 24	160	3
7	MBConv6, k5x5	24 × 24	272	4
8	MBConv6, k3x3	12 × 12	448	1
9	Conv1x1 & Pooling & FC	12 × 12	1792	1

EfficientNet-B4 model scaled up from the EfficientNet-B0 model by using the method of compound of compound scaling. This model is having more depth and its blocks are very deep.[18] Its compound scaling method comprises scaling in network (width, depth, and resolution) simultaneously with defined coefficients of scaling being the core of EfficientNet. The formula is:

$$\Phi = \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \tag{3}$$

Here, α, β, and γ are the scaling coefficients for depth, width, and resolution, respectively. Based on the baseline network EfficientNet-B0 (Φ=1), the authors determined α=1.2, β=1.1, and γ=1.15 through grid search. With these coefficients fixed, different Φ values are used to scale EfficientNet-B0 to derive a family of neural networks, EfficientNet-B1 to B7. EfficientNet-B4 corresponds to Φ=4. The total FLOPS increases by a factor of $(\alpha \cdot \beta 2 \cdot \gamma 2)\phi$ when scaling the network.

3.3.3. ConvNext-Tiny Model Architecture

ConvNeXt-Tiny is a lightweight convolutional neural network architecture proposed by Liu et al. in the paper *"A ConvNet for the 2020s"* [19] published in 2022 that aims to combine the strengths of modern Transformer-based models with the efficiency and inductive bias of traditional ConvNets. ConvNeXt-Tiny is part of the ConvNeXt family, which modernizes standard ConvNet designs by incorporating architectural improvements drawn from Vision Transformers (ViTs) while maintaining pure convolutional operations. ConvNeXt-Tiny adopts several key modifications over traditional ConvNets, especially ResNet-like architectures, including: Patchify Stem, Depthwise Separable Convolutions, Layer Normalization, GELU Activation Function, Inverted Bottlenecks, and Residual Connections. The detailed architecture is such as:

- **Stage 1:** The input image is 56 x 56 convolved with a stride of 4 producing 96 channels. It is first feature extraction stage.
- **Stage 2:** This stage contains 3 blocks. For each block, we have a 28×28 depthwise image. This stage has 192 channels.
  Stage 3: This stage contains 9 blocks, with the same structure as the blocks in Stage 2. The number of channels in this stage is 384.
- **Stage 4:** This stage contains 3 blocks, with the same structure as the blocks in Stage 2. There are total 768 numbering of channels present in this stage.
- **Classification Head:** The output of Stage 4 is processed by a global average pooling layer, followed by a fully connected layer to produce the final classification results. This stage has 1000 output channels. Each ConvNeXt block contains: Depthwise Convolution (7×7), LayerNorm, Pointwise convolution (1×1) for channel expansion and contraction, GELU activation, and Residual connection.

A typical ConvNeXt block operation can be expressed as:

$$\mathbf{B = A + PW2(GELU(PW1(LN(DepthwiseConv(A)))))} \tag{4}$$

where, DepthwiseConv(A) performs spatial feature extraction per channel, LN denotes Layer Normalization, PW1 and PW2 are pointwise (1×1) convolutions for dimensionality transformation, GELU is the non-linear activation, and the residual connection A + ... helps maintain the integrity of the input signal. This configuration results in a highly optimized, scalable, and performant convolutional network that competes closely with ViT models in accuracy while being more hardware-efficient. The Model Size is ~29M parameters. The FLOPs is ~4.5 GFLOPs. It has Top-1 Accuracy (ImageNet) of ~82.1%. ConvNeXT is a pure convolutional model (ConvNet), inspired by the design of Vision Transformers, that claims to outperform them. The authors started from a ResNet and "modernized" its design by taking the Swin Transformer as inspiration. Figure 5 represents the detailed architecture of ConvNext Model.

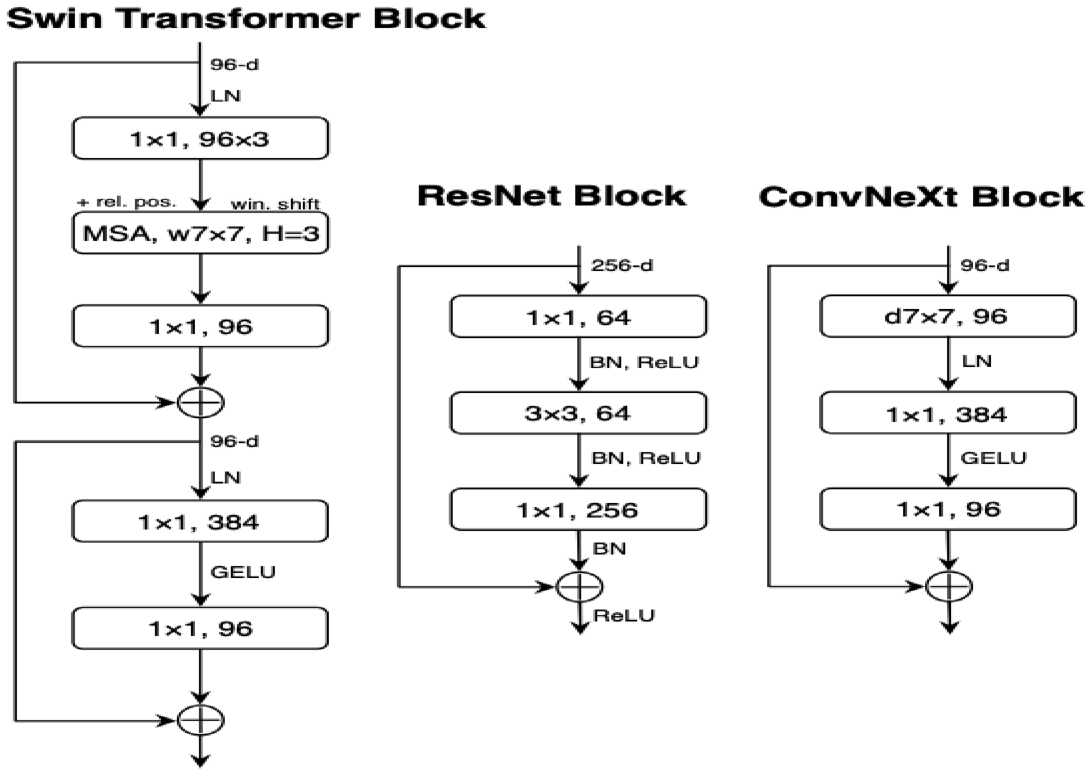


Figure 5. Detailed Architecture of ConvNext Model

3.3.4. Enhanced ConvNext Tiny Model Architecture

The Enhanced ConvNeXt-Tiny model is an optimized version of the original ConvNeXt-Tiny, initially proposed by Zhuang Liu et al. in their paper *"A ConvNet for the 2020s"* [20]. While the base ConvNeXt architecture was developed to modernize CNNs using design principles from Vision Transformers (ViTs), the enhanced version introduces domain-specific modifications aimed at improving performance in tasks such as pest classification. The Enhanced ConvNeXt-Tiny incorporates several architectural and training-based improvements over the base model: Incorporation of Squeeze-and-Excitation (SE) Modules**,** DropPath Regularization, Feature Pyramid Strategy, Optimized Normalization, and Domain-specific Pretraining (Optional). The detailed architecture is presented in Table 7.

Table 7. Stage-wise configuration of Enhanced Convnext-Tiny Model

Stage	Resolution	Block Type	#Blocks	Output Channels	Enhancements
Stem	224 × 224	Conv (4×4, stride=4)	1	96	-
Stage 1	56 × 56	ConvNeXt Block + SE	3	96	SE module, DropPath
Stage 2	28 × 28	ConvNeXt Block + SE	3	192	DropPath, InstanceNorm
Stage 3	14 × 14	ConvNeXt Block + SE	9	384	Feature Pyramid connection starts here

Stage 4	7 × 7	ConvNeXt Block + SE	3	768	Global feature fusion, DropPath
Head	7 × 7	GAP + FC	1	1000 (or num classes)	Domain-specific fine-tuning

The Enhanced ConvNeXt block can be described by the following equation:

$$\mathbf{B = A + DropPath(SE(PW2(GELU(PW1(LN(DWConv(A)))))))} \quad (5)$$

Where, DWConv is Depthwise Convolution with 7×7 kernel, LN is Layer Normalization, PW1, PW2 are Pointwise convolutions, SE is Squeeze-and-Excitation module, DropPath is Stochastic depth for regularization, and the output is added back with the input (X) via residual connection. If feature pyramids are applied, an additional lateral connection is computed:

$$\mathbf{P\_i = Conv1x1(F\_i) + Upsample(P\_\{i+1\})} \quad (6)$$

Where P_i denotes the feature map at level *i*, and F_i is the feature from the main block at that level. It has the Model Size of ~30–32M parameters (depending on SE and pyramid design). The FLOPs **is** ~5.1 GFLOPs.It has the top-1 Accuracy (ImageNet or domain-specific)**:** ~83%+.

3.3.5. Res-Net 50 Model Architecture

The ResNet-50 model was proposed by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their seminal paper titled *"Deep Residual Learning for Image Recognition"* [21] published in 2015. ResNet introduced the concept of residual learning to address the degradation problem in deep neural networks, where increasing depth leads to higher training error. The core innovation is the skip connection, which allows gradients to flow more easily during backpropagation and enables the training of very deep networks. ResNet-50 is one of the most widely used variants of the ResNet family, comprising 50 layers, structured using residual blocks with bottleneck design, which improves computational efficiency. The architecture of ResNet-50 can be split into different stages, each consisting of a specific number of Bottleneck Residual Blocks. Each block includes a series of 1×1, 3×3, and 1×1 convolutions and uses identity or projection shortcuts. Conv1 is the initial convolution layer with a 7×7 kernel followed by max pooling. Conv2_x to Conv5_x are residual block stages that progressively increase the number of filters while reducing spatial resolution. Average Pooling is global average pooling after the final block. Fully Connected Layer is the dense layer with 1000 units (can be adapted for a specific number of classes). These are illustrated below in Table 8:

Table 8. Layered Structure of Res-Net 50 Model

Stage	Operator	Output Size	Number of Blocks	Number of Filters
Conv1	7×7, 64, stride 2	112×112	1	64
	3×3 Max Pool, stride 2	56×56	-	-
Conv2_x	[1×1, 64], [3×3, 64], [1×1, 256] × 3	56×56	3	64 → 256
Conv3_x	[1×1, 128], [3×3, 128], [1×1, 512] × 4	28×28	4	128 → 512

Conv4_x	[1×1, 256], [3×3, 256], [1×1, 1024] × 6	14×14	6	256 → 1024
Conv5_x	[1×1, 512], [3×3, 512], [1×1, 2048] × 3	7×7	3	512 → 2048
AvgPool	Global Average Pooling	1×1	-	-
FC	Fully Connected (Dense)	1×1	1	1000 (or custom classes)

The residual block allows the model to learn the residual mapping rather than the full transformation, which simplifies optimization. A residual block is expressed as:

$$\mathbf{b = F(a,\{W_i\}) + b} \tag{7}$$

where, a is the input to the block, $F(x,\{W_i\})$ is the residual function to be learned (typically two or three stacked convolutional layers), and b is the output of the block.

If the input and output dimensions differ, a linear projection W_s is applied to match dimensions:

$$\mathbf{B = F(a,\{W_i\}) + W_s a} \tag{8}$$

This identity mapping enables better gradient flow and makes training deeper networks more feasible.

3.4. Performance Measures

It is very important to monitor the performance of deep learning models. It enhances their efficiency in correctly identifying and classifying peanut pests. In our research work, we utilized several standard classification metrics derived from the confusion matrix to evaluate the performance of the five models used: ConvNeXt-Base, ConvNeXt-Tiny, Enhanced ConvNeXt-Tiny, EfficientNet-B4, and ResNet-50. These metrics include Accuracy, Precision, Recall, F1-Score, and Confusion Matrix, which are widely accepted evaluation criteria in deep learning classification tasks [22].

3.4.1. Confusion Matrix

A summary of prediction results on a classification problem is shown in a confusion matrix. It presents the number of true and false predictions broken down by each class. The given Table 9 illustrates this:

Table 9. Layered Structure of Res-Net 50 Model

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

3.4.2. Accuracy

Accuracy is the term that how accurate the results are given by model. The ratio of correctly predicted outcomes to the total number of predictions is known as accuracy. It gives an overall effectiveness of the model. This expression can be expressed in fraction as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

3.4.3. Precision

Precision refers to the ratio of correctly predicted positive observations to the total predicted positive High precision shows low false positive rate. This can be presented as:

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

3.4.4. Recall

Recall is the ratio of correctly predicted positive observations to all actual positive observations. High recall indicates that the model is capturing most of the relevant instances. It can be shown as:

$$R = \frac{TP}{TP + FN} \tag{11}$$

3.4.5. F1-Score

The F1-Score is the weighted average of Precision and Recall. It is especially useful when dealing with imbalanced datasets. The equation to show F1-Score is as follows:

$$F = 2 \times \frac{precision \times recall}{precision + recall} \tag{12}$$

3.4.6. Model Evaluation Step

All the models were trained and evaluated on a balanced pest image dataset. The performance was assessed using 5-fold cross-validation to ensure generalizability and reduce overfitting. The metrics were calculated for each fold and averaged to produce the final results.

4. Results and Discussions

A detailed analysis of the results is given below which is obtained by using five models: ConvNeXt-Base, ConvNeXt-Tiny, Enhanced ConvNeXt-Tiny, EfficientNet-B4, and ResNet-50. All experiments were conducted using deep learning framework. Training and evaluation were performed on a GPU-enabled system. The dataset was split into training (80%), validation (10%), and testing (10%) subsets for each pest class. The models were fine-tuned using transfer learning, where ImageNet-pretrained weights were utilized. The models were evaluated using various parameters. Each parameter such as accuracy, precision, recall, and F-measure is discussed below in this section.

4.1. Results obtained by using ConvNeXt-Base

With an accuracy of 98.54%, ConvNeXt-Base is azure by an overall precision of 98.38%, but lacks recall at 96.21% and F1-score of 92.17%.

4.1.1. Classification report metrics of ConvNext-Base

The classification report metrics of the ConvNext-Base model across three pest classes such as Aphids, Caterpillar, and Wireworm, is shown below in figure 6. The model achieved nearly perfect scores, with precision, recall, and F1-score all close to 100% for Aphids, indicating highly accurate classification. For Caterpillar, the model maintained a high precision of 100%, which illustrates that almost all images predicted as Caterpillar were indeed correct; however, the recall decreased to around 89%. Due to this the F1-score diminished to approximately 94%, suggesting minor inconsistencies in identifying all positive samples. In the case of Wireworm, the model again showed excellent balance with a precision of 96%, a perfect recall of 100%, and a strong F1-score of about 98%. This indicates that the model retrieved all Wireworm instances accurately and made very few false predictions. Overall, these metrics confirm the reliability and robustness of the ConvNeXt-Base model in distinguishing pest types, specifically excelling in reducing false positives and false negatives across most classes.

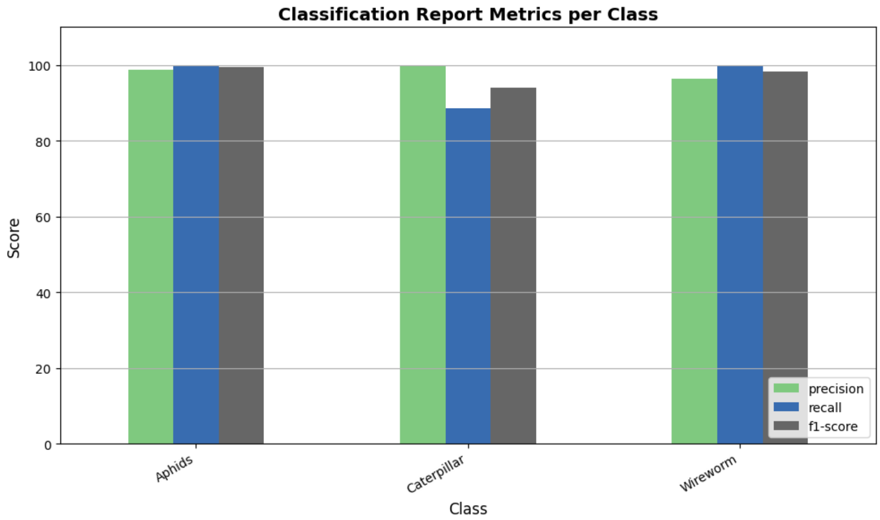


Figure 6. Classification report using ConvNext-Base Model

4.1.2. Confusion matrix using ConvNext-Base

Figure 7 shows further challenges. The confusion matrix gives a clear summary of the classification performance of the ConvNeXt-Base model across three pest classes: Aphids, Caterpillar, and Wireworm. The diagonal elements illustrate the number of correctly classified instances for each class, while off-diagonal elements show misclassifications. Aphids were perfectly recognized with 245 out of 245 images correctly predicted, showing no misclassifications into any other category. For Caterpillar, 39 out of 44 total true instances, were correctly classified. However, 3 were misclassified as Aphids, and 2 were misclassified as Wireworm. Wireworm classification was also flawless, with all 53 instances accurately identified and no confusion with other classes. This indicates a very strong distinction of Wireworm features by the model.

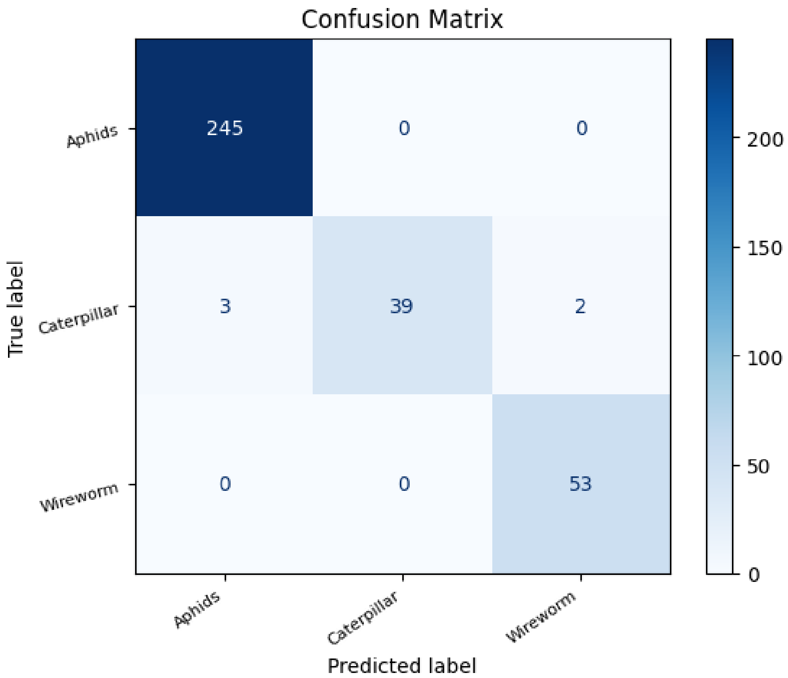


Figure 7. Confusion Matrix report using ConvNext-Base Model

4.1.3. Loss dynamics related to ConvNext-Base

As shown in Figure 8, there is a continuous decline in training loss up to the 16th epoch, and it converges at around 0.50. The fact that it is effective learning during training. But, although the validation loss (not shown) is slightly up after epoch 4, it indicates a small amount of over fitting.

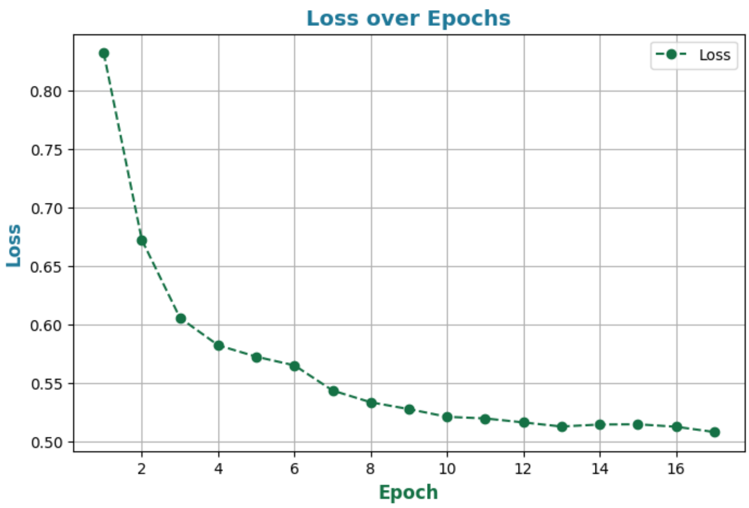


Figure 8. Loss Dynamics report using ConvNext-Base Model

4.1.4 Training vs. value accuracy using ConvNext-Base

The accuracy curves in Figure 9 show 98.54% training accuracy and 96.21% validation accuracy. This will confirm that overfitting, since the model memorizes training patterns and generalizes bad on validation data.

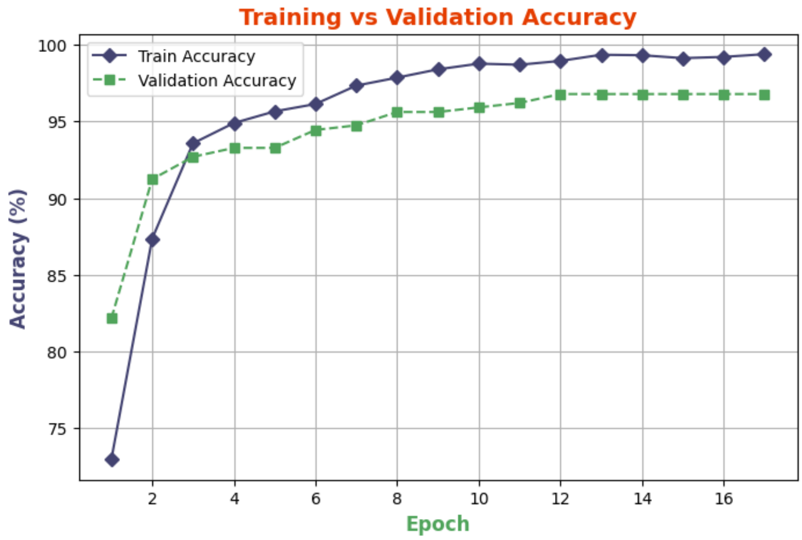


Figure 9. Training vs Validation Accuracy Graph Using Convnext-Base Model

4.2. *Results obtained by using ConvNext-Tiny*

The ConvNeXt-Tiny model showed the exceptional results in our research work and proved to be the best among other modernized models applied on the same publicly available dataset. The highest accuracy shown in our work is 99.42%, by using the ConvNeXt-Tiny model, with overall precision, recall, and F1-score of 99.11%.

4.2.1. Classification report metrics of ConvNext-Tiny

The classification report clearly shows that various classes performed well using the ConvNeXt-Tiny model. Figure x demonstrates the results of all the classes, including Aphids, Caterpillar, and wireworm using the ConvNeXt-Tiny. Figure 10 below represents the classification report.

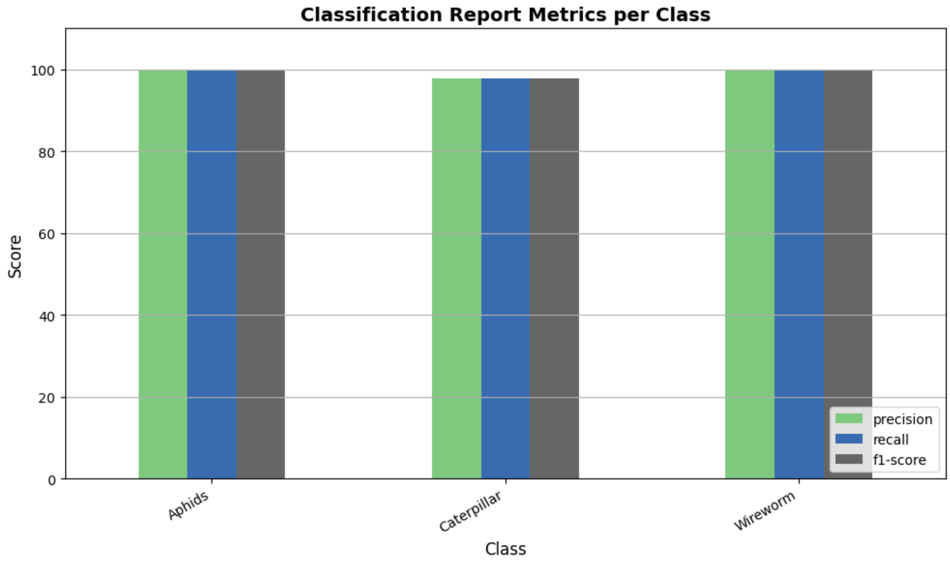


Figure 10. Classification Report Using Convnext-Tiny Model

4.2.2. Confusion matrix using ConvNext-Tiny

Visual representation of all the classes present in our dataset is shown in Figure 11 which is representing the confusion matrix of all the peanut image classes using the ConvNeXt-Tiny model.

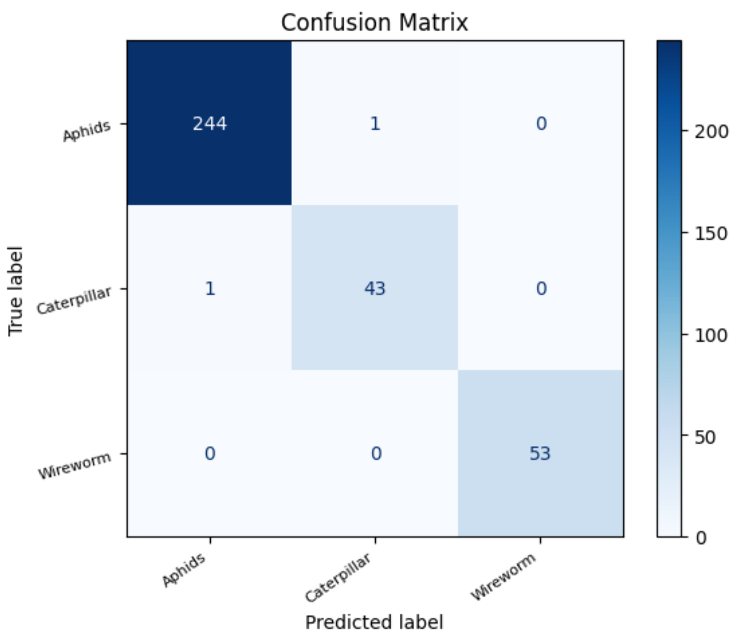


Figure 11. Confusion Matrix report using Convnext-Tiny Model

4.2.3. Loss dynamics related to ConvNext-Tiny

Loss dynamics are shown in the figure x which appeared by using the ConvNeXt-Tiny model in our work. Loss dynamics graph is given below in figure 12.

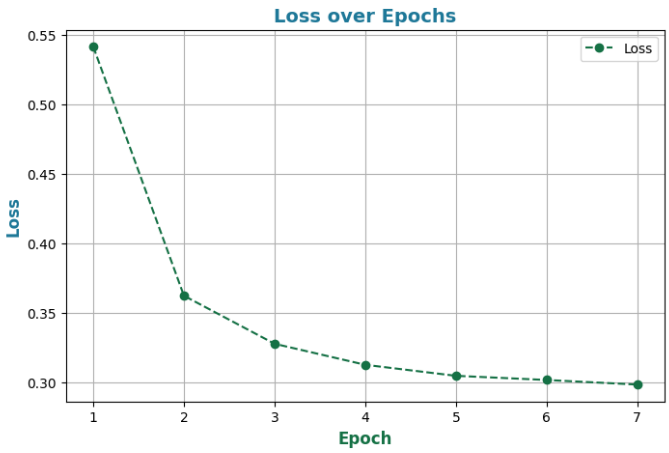


Figure 12. Loss Dynamics report using Convnext-Tiny Model

4.2.4. Training vs value accuracy using ConvNext-Tiny

By using the ConvNeXt-Tiny model in our work, we obtained a validation and training accuracy nearly 100%. Figure x represents a visual view of training versus validation accuracy.

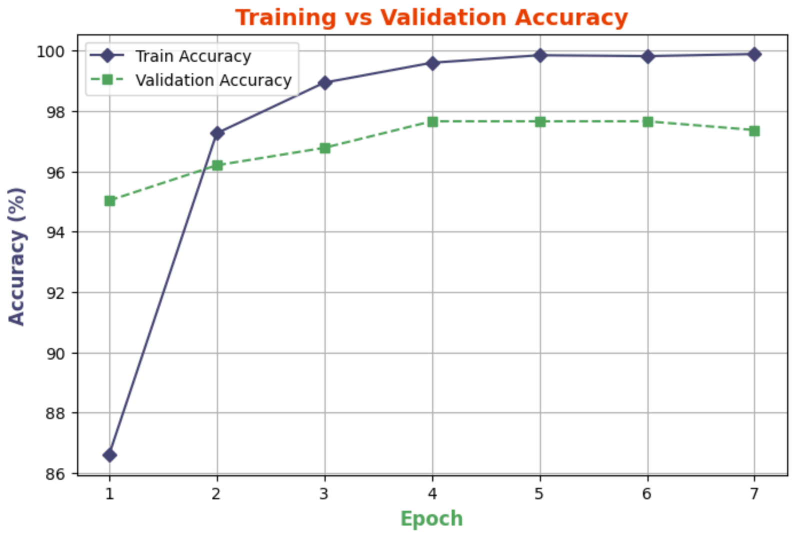


Figure 13. Training vs Validation Accuracy Graph using Convnext-Tiny Model

4.3. Results obtained by using enhanced ConvNext-Tiny

We obtained an accuracy of 98.54% and a precision of 97.28%. While Recall and F1-score were 97.46% and 97.33%, respectively.

4.3.1. Classification report metrics of Enhanced Convnext-Tiny

The classification report metrics of Enhanced ConvNeXt-Tiny model across three pest classes such as Aphids, Caterpillar, and Wireworm is shown below in figure 14. The model achieved nearly perfect scores, with precision, recall, and F1-score all close to 100% for Aphids, indicating highly accurate classification. For Caterpillar, the model maintained a high precision nearly 100%, which illustrates that almost all images predicted as Caterpillar were indeed correct; however, the recall decreased to around 92%. Due to this the F1-score diminished to approximately 95%, suggesting minor inconsistencies in identifying all positive samples. In the case of Wireworm, the model showed a precision of nearly 95%, a perfect recall of 100%, and a strong F1-score of about 98%. This indicates that the model retrieved all Wireworm instances accurately and made very few false predictions. Overall, these metrics confirm the reliability and robustness of the ConvNeXt-Base model in distinguishing pest types, specifically excelling in reducing false positives and false negatives across most classes.

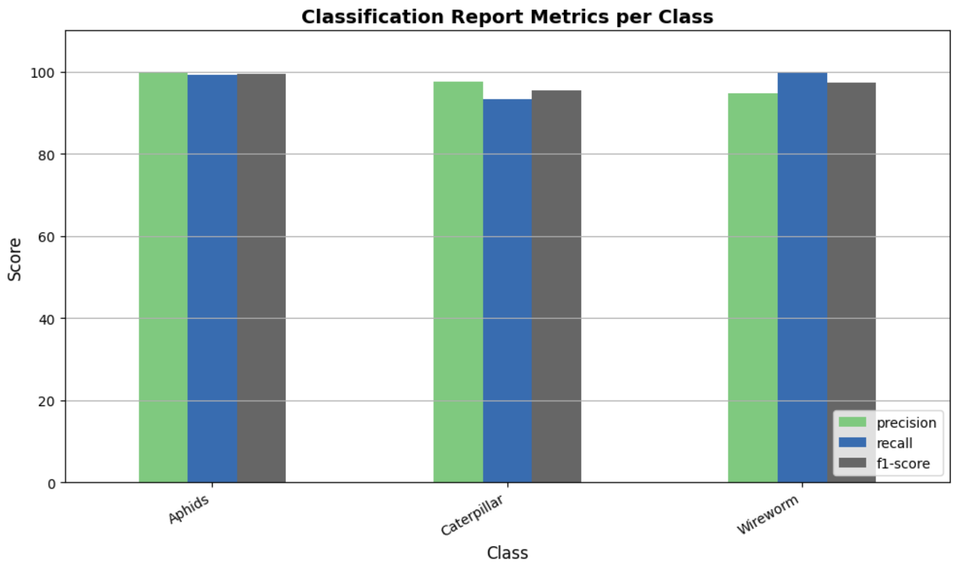


Figure 14. Classification Report Using Enhanced Convnext-Tiny Model

4.3.2**.** Confusion matrix using Enhanced ConvNeXt-Tiny

Figure 15 below shows further challenges. The confusion matrix gives a clear summary of the classification performance of the Enhanced ConvNeXt-Tiny model across three pest classes: Aphids, Caterpillar, and Wireworm. The diagonal elements illustrate the number of correctly classified instances for each class, while off-diagonal elements show misclassifications. Aphids were perfectly recognized with 243 out of 245 images correctly predicted, showing less misclassifications into any other category. For Caterpillar, 41 out of 44 total true instances were correctly classified. However, 1 was misclassified as Aphids, and 2 were misclassified as Wireworm. Wireworm classification was also flawless, with all 53 instances accurately identified and no confusion with other classes. This indicates a very strong distinction of Wireworm features by the model.

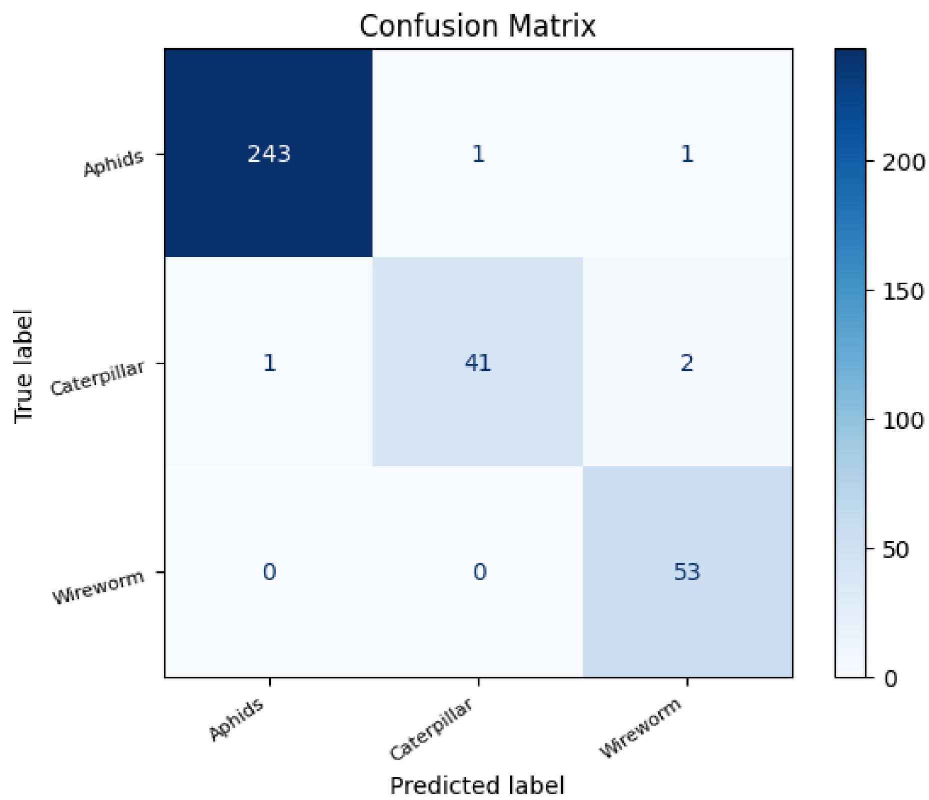


Figure 15. Confusion Matrix Report Using Enhanced Convnext-Tiny Model

4.3.3. Loss dynamics related to Enhanced ConvNeXt-Tiny

As shown in Figure 16, there is a continuous decline in training loss up to the 4.5 epoch, and it converges at around 0.60. Then it rise again till 6 epoch, converges at nearly 0.62.After that, it decreased again till 10.5 epoch, and increased again till 12 epoch. It starts to drop again after 12 epochs.

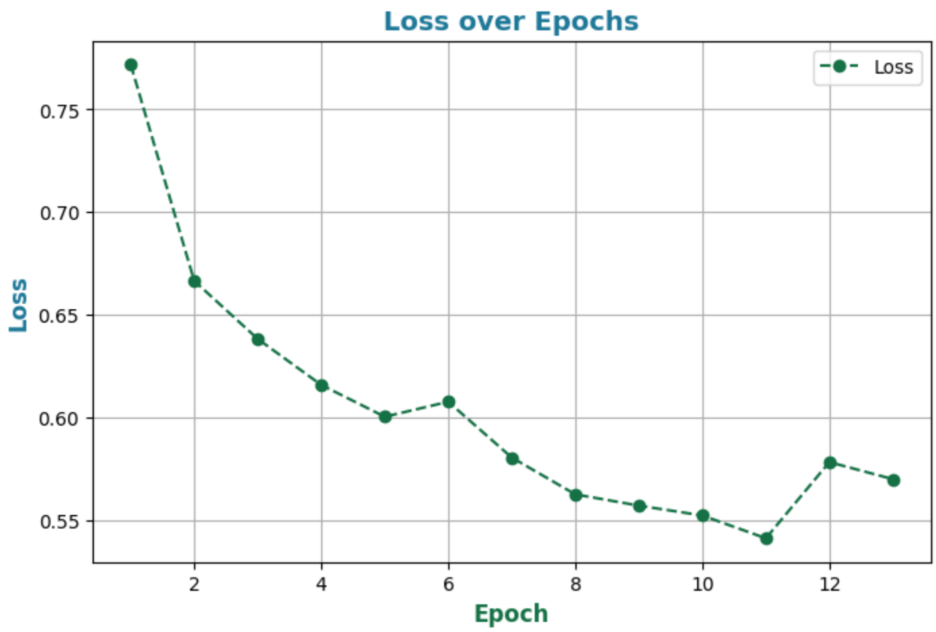


Figure 16. Loss Dynamics report using Enhanced Convnext-Tiny Model

4.3.4. Training vs value accuracy using Enhanced ConvNeXt-Tiny

By using the Enhanced ConvNeXt-Tiny model in our work, we obtained a validation and training accuracy as illustrated in the graph clearly. Figure 17 represents a visual view of training versus validation accuracy.

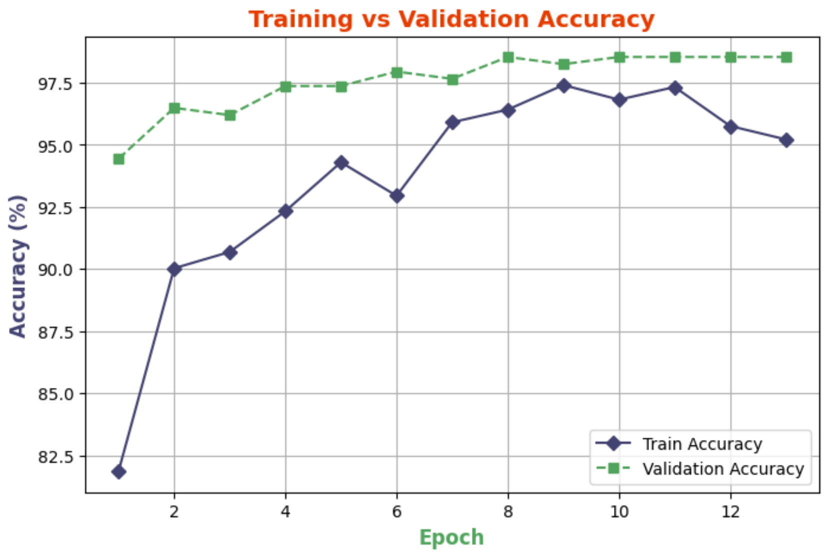


Figure 17. Training vs Validation Accuracy Graph using Enhanced Convnext-Tiny Model

4.4. Results obtained by using the EfficientNet_B4 Model

Accuracy shown by the EfficientNet_B4 model is 94.44%. The values of other parameters such as precision, recall, and f1-score are 90.26%, 92.57%, and 91.37% respectively.

4.4.1. Classification report metrics of EfficientNet-B4 model

The classification report metrics of the EfficientNet-B4 model across three pest classes such as Aphids, Caterpillar, and Wireworm is shown below in figure 18. The model achieved nearly perfect scores, with precision, recall, and F1-score all close to 100% for Aphid. For Caterpillar, the model maintained a low precision nearly 80%, which illustrates that mostlty images predicted as Caterpillar were incorrect; however, the recall decreased to around 82%. Due to this the F1-score diminished to approximately 81%, suggesting major inconsistencies in identifying all positive samples. In the case of Wireworm, the model showed a precision of nearly 95%, a recall of neary 99%, and a strong F1-score of about 97%.

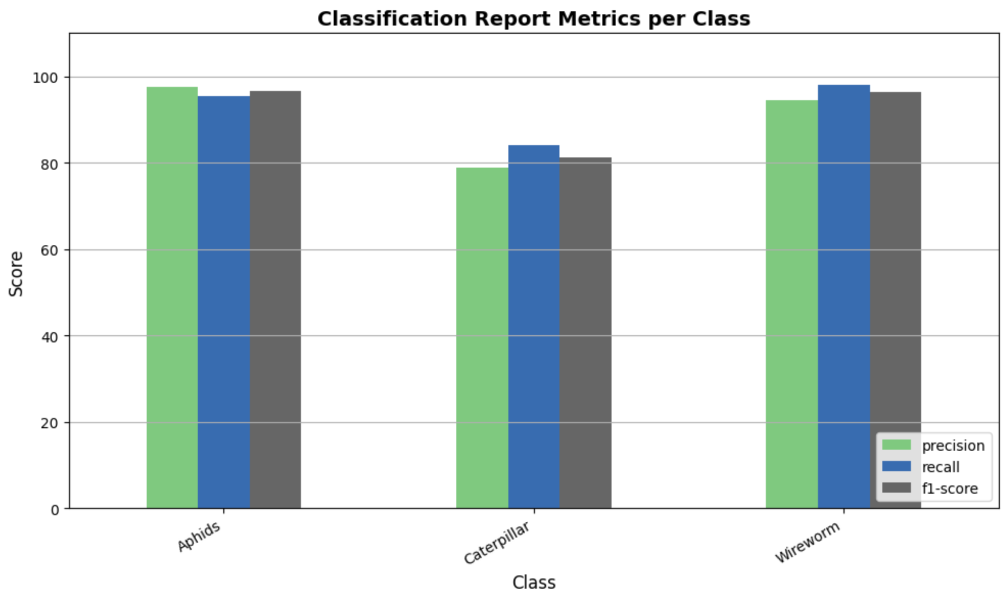


Figure 18. Classification report using EfficientNet-B4 Model

4.4.2. Confusion matrix using EfficientNet-B4 Model

The confusion matrix gives a clear summary of the classification performance of the EfficientNEt-B4 model across three pest classes: Aphids, Caterpillar, and Wireworm. The diagonal elements illustrate the number of correctly classified instances for each class, while off-diagonal elements show misclassifications. Aphids were perfectly recognized with 234 out of 245 images correctly predicted, showing less misclassifications into any other category. For Caterpillar, 37 out of 44 total true instances, were correctly classified. However, 5 were misclassified as Aphids, and 2 were misclassified as wireworms. Wireworm classification was almost flawless, with all 52 instances accurately identified and 1 confusion with Aphids class. Figure 19 below shows further challenges.

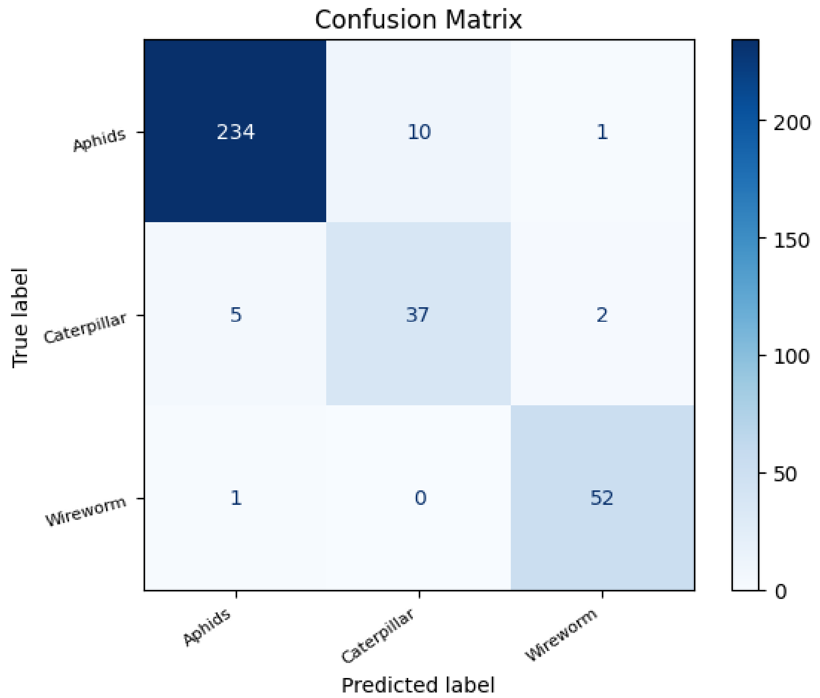


Figure 19. Confusion Matrix Report Using EfficientNet-B4 Model

4.4.3. Loss dynamics related to the EfficientNet-B4 Model

As shown in Figure 20, there is a continuous decline in training loss up to the 2 epoch, and it converges at around 0.18. Then it decreases again linearly till 5 epoch.

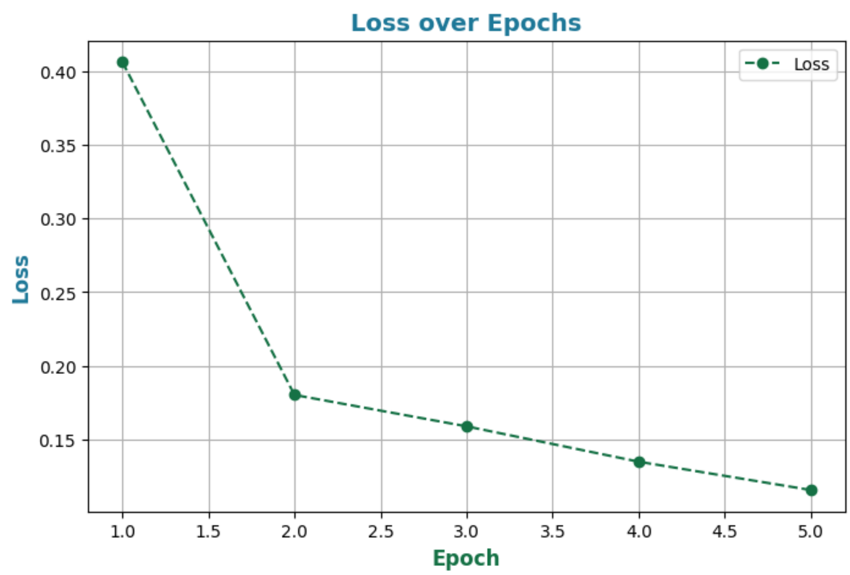


Figure 20. Loss Dynamics report using EfficientNet-B4 Model

4.4.4. Training vs value accuracy using EfficientNet-B4 Model

By using the EfficientNet-B4 model in our work, we obtained a validation and training accuracy as illustrated in the graph clearly. Figure 21 represents a visual view of training versus validation accuracy.

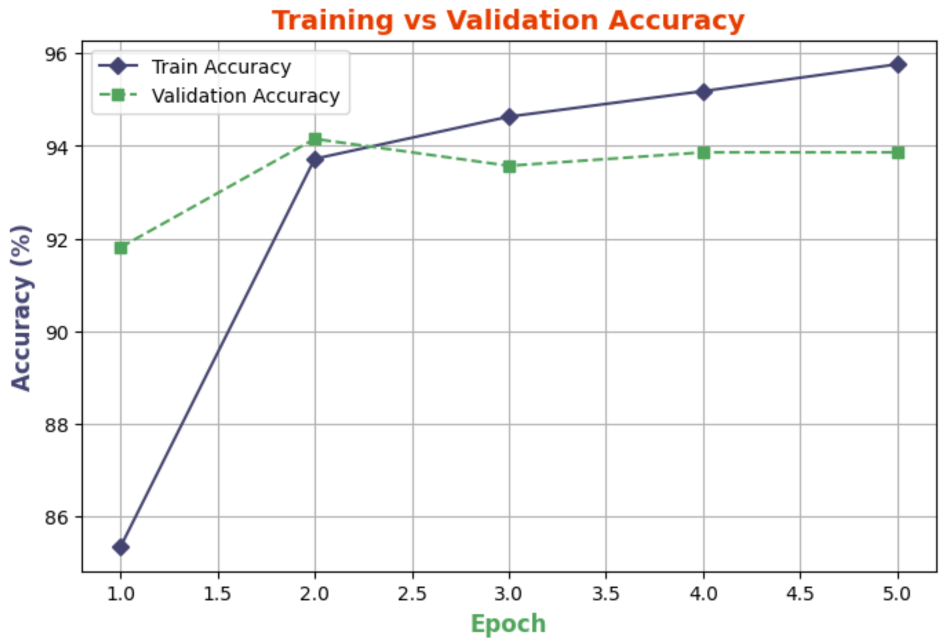


Figure 21. Training vs Validation Accuracy Graph using EfficientNet-B4 Model

4.5. Results obtained by using the Resnet-50 Model

Accuracy shown by ResNet-50 MODEL is 97.08%. The values of other parameters such as precision, recall, and f1-score are 96.49%, 93.05%, and 94.44% respectively.

4.5.1. Classification report metrics of the ResNet-50 model

The classification report metrics of the ResNet-50 model across three pest classes such as Aphids, Caterpillar, and Wireworm is shown below in figure 22. The model achieved nearly perfect scores, with precision, recall, and F1-score all close to 100% for Aphids and wireworm. However for Caterpillar, the model maintained a precision nearly 95%, which illustrates that mostlty images predicted as Caterpillar were correct; however, the recall decreased to 80%. Due to this, the F1-score diminished to approximately 83%, suggesting major inconsistencies in identifying all positive samples.

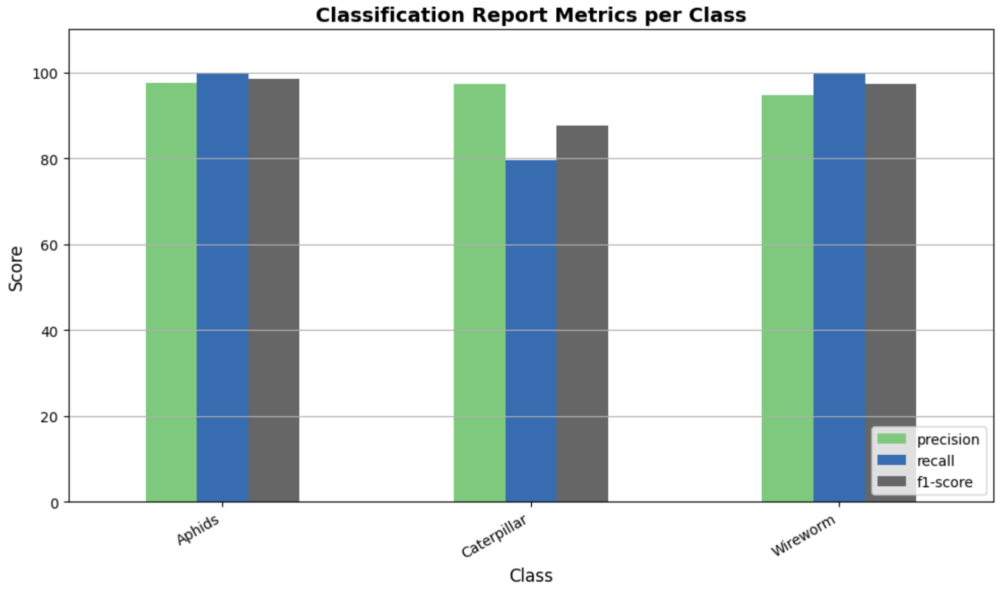


Figure 22. Classification Report Using ResNet-50 Model

4.5.2. Confusion matrix using ResNet-50 Model

The confusion matrix gives a clear summary of the classification performance of the ResNet-50 model across three pest classes: Aphids, Caterpillar, and Wireworm. The diagonal elements illustrate the number of correctly classified instances for each class, while off-diagonal elements show misclassifications. Aphids were perfectly recognized with 244 out of 245 images correctly predicted, showing less misclassifications into any other category. For Caterpillar, 35 out of 44 total true instances, were correctly classified. However, 6 were misclassified as Aphids, and 3 were misclassified as wireworms. Wireworm classification was almost flawless, with all 53 instances accurately identified and no confusion with other classes. Figure 23 below shows all the illustrations related to this.

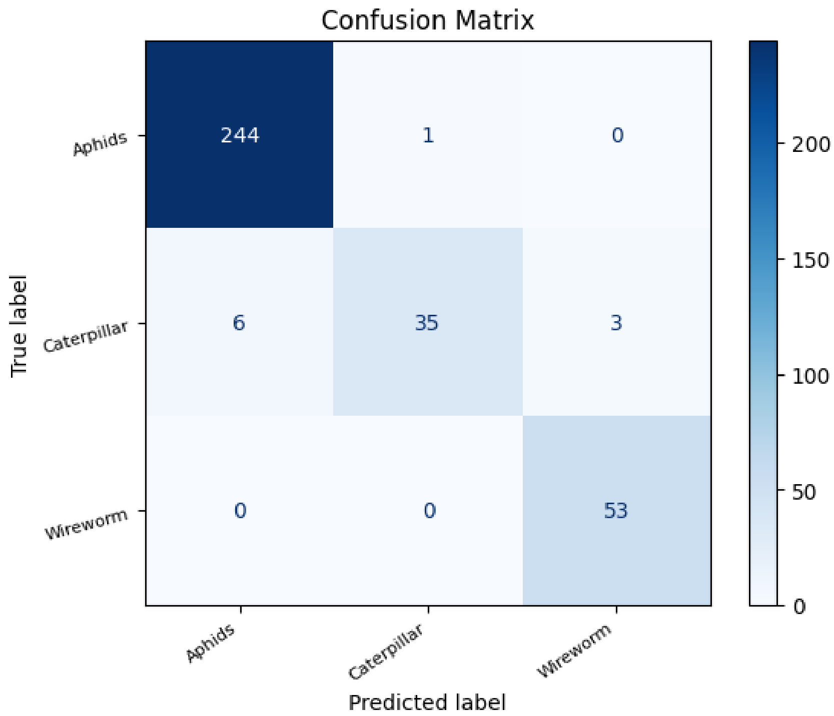


Figure 23. Confusion Matrix Report Using ResNet-50 Model

4.5.3. Loss dynamics related to the ResNet-50 Model

There is a continuous decline in training loss up to the 2 epoch, and it converges at around 0.63. Then it decreases again linearly till 12 epochs. These details are illustrated in Figure 24 below.

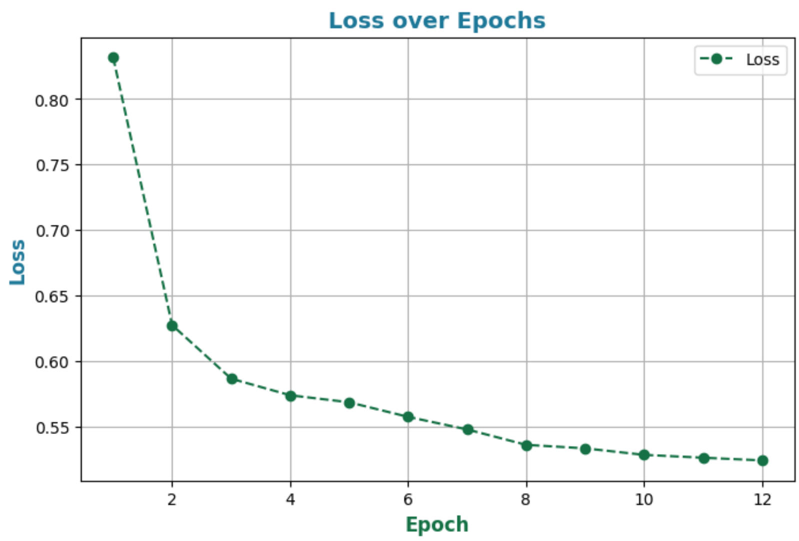


Figure 24. Loss Dynamics report using ResNet-50 Model

4.5.4. Training vs value accuracy using ResNet-50 Model

By using the ResNet-50 model in our work, we obtained a validation and training accuracy as illustrated in the graph. Figure 25 represents a visual view of training versus validation accuracy.

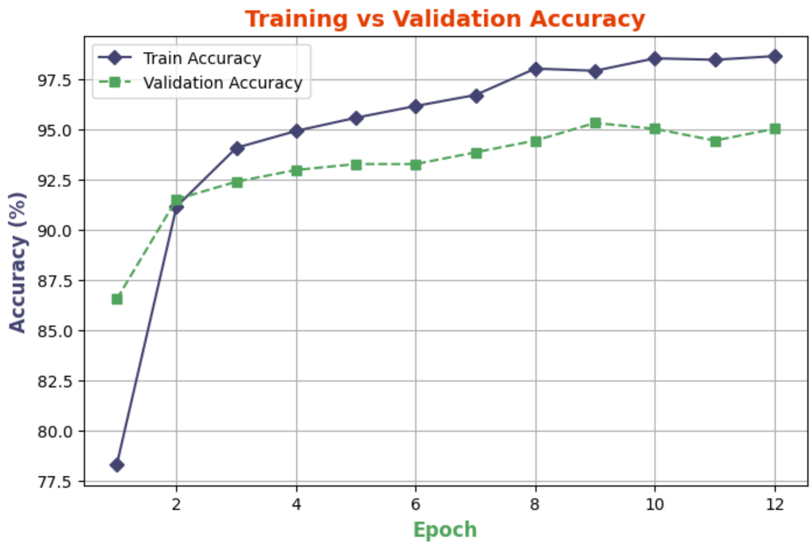


Figure 25. Training vs Validation Accuracy Graph using RestNet-50 Model

4.6. Results obtained by using the ConvNext-Tiny Model by using five classes (self-created dataset)

Accuracy obtained by ConvNext-Tiny model is 91.94%. The values of other parameters such as precision, recall, and f1-score are 88.27%, 87.83%, and 87.93% respectively.

4.6.1. Classification report metrics of ConvNext-Tiny Model

The classification report metrics of the ConvNext-Tiny model across five pest classes such as Aphids, Caterpillar, Wireworm, Thrips and Armyworm, is shown below in figure 26. The model achieved nearly perfect scores, with precision, recall, and F1-score all close to 97% for Aphids and wireworm. However for Armyworm, the model maintained a precision nearly 95%, which illustrates that most images predicted as Armyworm were correct; however, the recall decreased to 81%. Due to this the F1-score also diminished to approximately 83%, suggesting major inconsistencies in identifying all positive samples. However for Caterpillar and Thrips, precision, recall and F1-score were almost 80%.

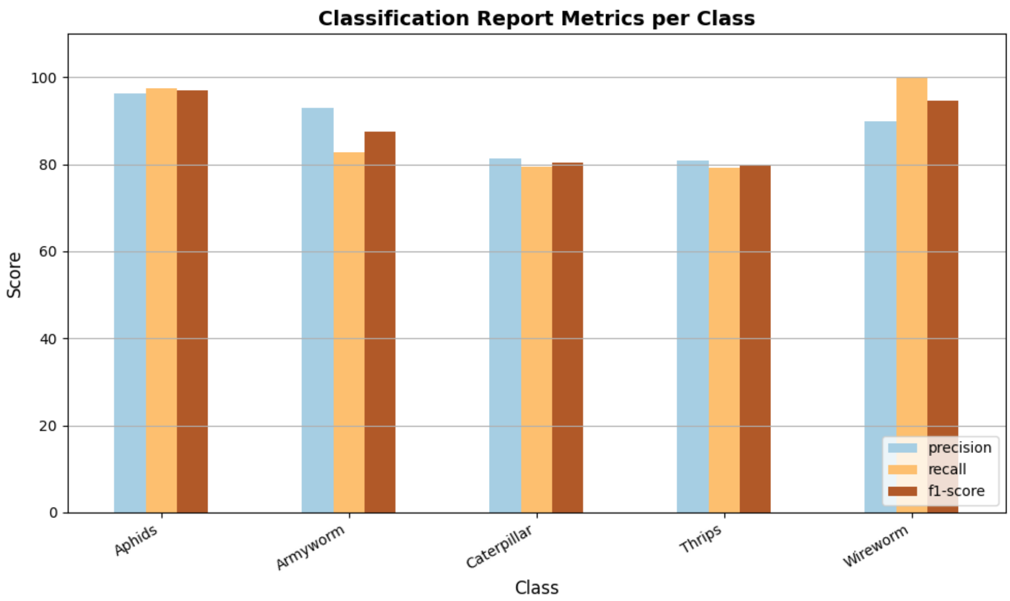


Figure 26. Classification Report using ConvNext-Tiny Model

4.6.2. Confusion matrix using ConvNext-Tiny Model

The confusion matrix gives a clear summary of the classification performance of the Convnext-TINY model across five pest classes. The diagonal elements illustrate the number of correctly classified instances for each class, while off-diagonal elements show misclassifications. Aphids were perfectly recognized with 239 out of 245 images correctly predicted, showing less misclassifications into any other category. For Caterpillar, 35 out of 44 total true instances, were correctly classified. Wireworm classification was flawless, with all 53 cases accurately identified and no confusion with other classes. Armyworm was recognized 53 out of 65. While Thrips were identified with 42 out of 52. Figure 27 below shows all the illustrations related to this.

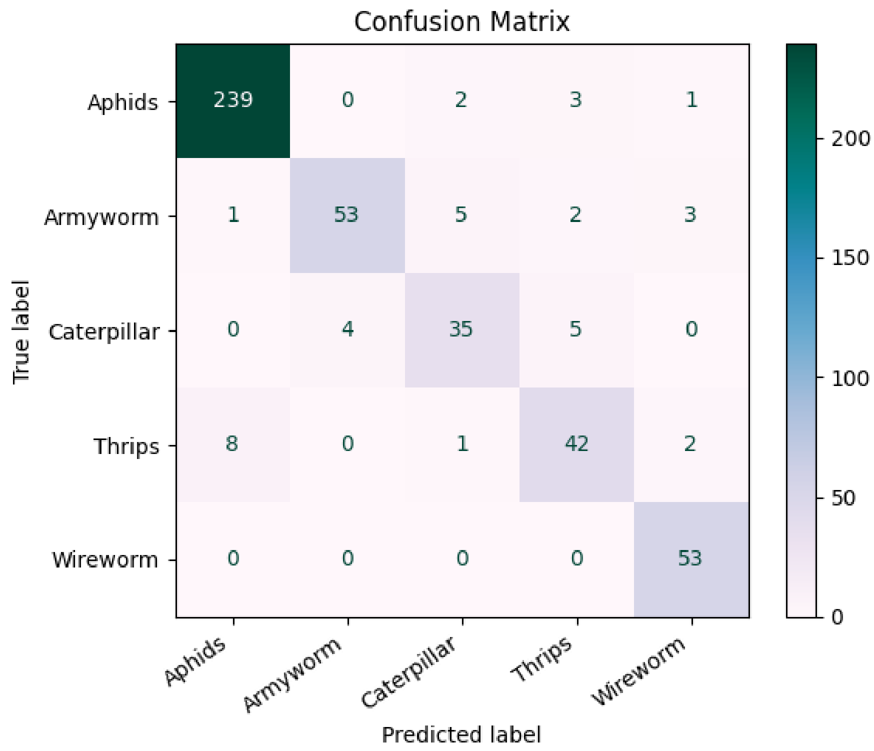


Figure 27. Confusion Matrix Report Using ConvNext-Tiny Model

4.6.3. Loss dynamics related to ConvNext-Tiny Model

There is a linear decline in training loss up to the 2 epoch, and it converges at around 0.62. Then it decreases again steadily till 9 epoch. These details are illustrated in the figure 28 below.

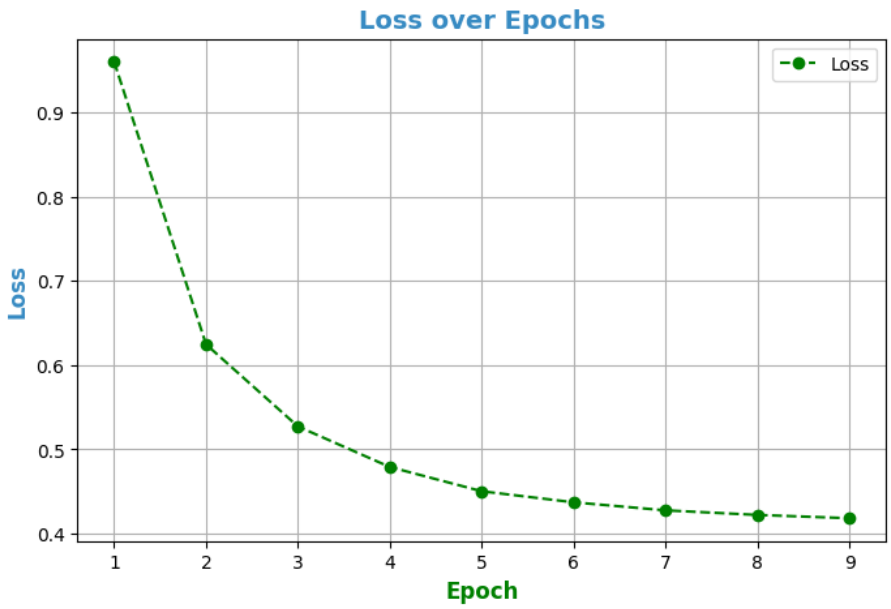


Figure 28. Loss Dynamics report using ConvNext-Tiny Model

4.6.4. Training vs value accuracy using ConvNext-Tiny Model

We obtained a validation and training accuracy using the ConvNext-Tiny model in our work. Figure x represents a visual view of training versus validation accuracy.

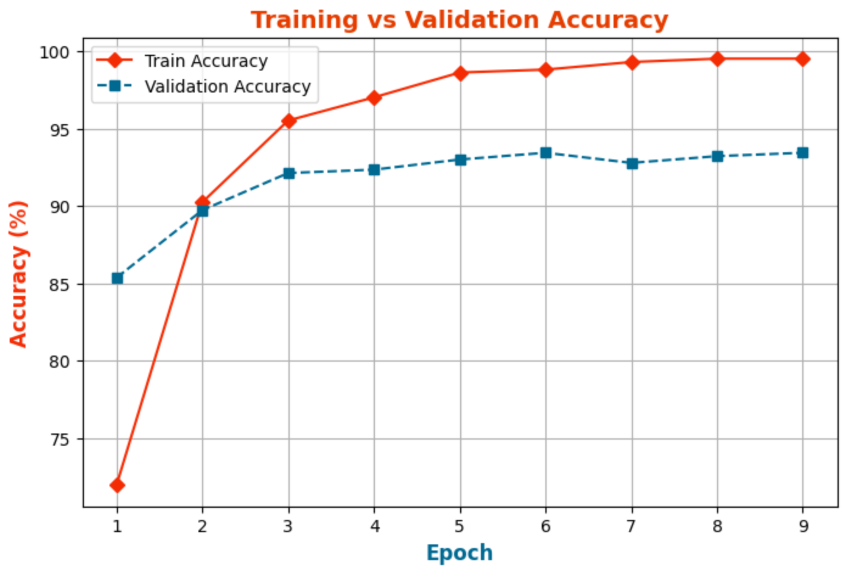


Figure 29. Training vs Validation Accuracy Graph using ConvNext-Tiny Model

4.7. Comparative view of all the performance measures using different models

In the table x, we'll be showing the results obtained related to different performance measures such as Accuracy, Precision, Recall, and F1-score using different models as ConvNeXt-Base, ConvNeXt-Tiny, Enhanced ConvNeXt-Tiny, EfficientNet-B4, and ResNet-50 in our work. The given Table 10 below demonstrates the comparative analysis of each model.

Table 10. Comparison of Performance Measures of each Model

Model Name	Accuracy	Precision	F1-Score	Recall
ConvNeXt-Base	98.54%	98.38%	97.17%	96.21%
ConvNeXt-Tiny	99.42%	99.11%	99.11%	99.11%
Enhanced ConvNeXt-Tiny	98.54%	97.28%	97.33%	97.46%
EfficientNet-B4	94.44%	90.26%	91.37%	92.57%
ResNet-50	97.08%	96.49%	94.44%	93.05%

5. CONCLUSION AND FUTURE WORK

We have explored and evaluated the performance of various deep learning-based models such as ConvNeXt-Base, ConvNeXt-Tiny, Enhanced ConvNeXt-Tiny, EfficientNet-B4, and ResNet-50, for the task of peanut pest identification using image classification. We utilized five pest categories including Aphids, Caterpillar, Wireworm, Armyworm, and Thrips. We observed that the ConvNeXt-Tiny model had the best results over others in terms of accuracy, precision, recall, and F1-score. Visualizations such as the classification report graph and confusion matrix further validated the model's robustness, especially in minimizing false positives and negatives. While most models demonstrated strong generalization capabilities, slight misclassifications were noted, especially in distinguishing Caterpillar from other similar pests, highlighting areas for potential improvement in feature discrimination. Our research demonstrates the practical viability of deep learning techniques in automating pest detection in peanut crops, improving sustainable agriculture, and reducing labor costs.

For future work, there are several things need to be improved. Expanding the dataset to include a large variety of pests would improve the generalization and scalability of the models. Future work may focus on integrating the trained model into a mobile or edge device application for real-time pest detection in orchards. More advanced data augmentation strategies can be used, which enhance model robustness under varying lighting and environmental conditions. Combining image data with environmental factors, e.g., temperature, humidity or pest lifecycle metadata, may lead to a more holistic pest prediction system. Through these enhancements, we aim to develop a more accurate, scalable, and interpretable solution that supports farmers and agricultural experts in timely pest identification and management.

References

[1] N. E. Chalabi, A. Attia, and A. Bouziane, "MULTIMODAL FINGER DORSAL KNUCKLE MAJOR AND MINOR PRINT RECOGNITION SYSTEM BASED ON PCANET DEEP LEARNING," *ICTACT Journal on Image and Video Processing*, vol. 10, no. 3, pp. 2153–2158, Feb. 2020, doi: 10.21917/ijivp.2020.0308.

[2] A. Abade, P. A. Ferreira, and F. de Barros Vidal, "Plant diseases recognition on images using convolutional neural networks: A systematic review," *Comput Electron Agric*, vol. 185, p. 106125, Jun. 2021, doi: 10.1016/j.compag.2021.106125.

[3] B. P. Battemarco *et al.*, "Water dynamics and blue-green infrastructure (BGI): Towards risk management and strategic spatial planning guidelines," *J Clean Prod*, vol. 333, p. 129993, Jan. 2022, doi: 10.1016/j.jclepro.2021.129993.

[4] Y. Tang, L. Gu, and L. Wang, "Deep Stacking Network for Intrusion Detection," *Sensors*, vol. 22, no. 1, p. 25, Dec. 2021, doi: 10.3390/s22010025.

[5] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, "IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2019, pp. 8779–8788. doi: 10.1109/CVPR.2019.00899.

[6] V. P. and I. M., "GNViT- An enhanced image-based groundnut pest classification using Vision Transformer (ViT) model," *PLoS One*, vol. 19, no. 3, p. e0301174, Mar. 2024, doi: 10.1371/journal.pone.0301174.

[7] P. Thakare and R. S. V., "Advanced pest detection strategy using hybrid optimization tuned deep convolutional neural network," *Journal of Engineering, Design and Technology*, vol. 22, no. 3, pp. 645–678, Apr. 2024, doi: 10.1108/JEDT-09-2021-0488.

[8] S. Dev, "Application of CNN Architecture for Insect Identification and Labeling in Peanut Crops," in *2023 International Conference on Data Science and Network Security (ICDSNS)*, IEEE, Jul. 2023, pp. 1–8. doi: 10.1109/ICDSNS58469.2023.10244877.

[9] S. S, C. R. Rathish, Y. K. M, S. R, S. R. K, and E. Vanaja, "Pest Detection using Machine Learning," in *2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS)*, IEEE, Apr. 2024, pp. 87–91. doi: 10.1109/ICC-ROBINS60238.2024.10533899.

[10] Y. Yun *et al.*, "Research on a Method for Identification of Peanut Pests and Diseases Based on a Lightweight LSCDNet Model," *Phytopathology*, vol. 114, no. 9, pp. 2162–2175, Sep. 2024, doi: 10.1094/PHYTO-01-24-0013-R.

[11] V. P and I. M., "GROUNDNUT CROP PEST DETECTION AND CLASSIFICATION USING COMPREHENSIVE DEEP-LEARNING MODELS," *Suranaree Journal of Science and Technology*, vol. 31, no. 1, pp. 020028(1–17), Feb. 2024, doi: 10.55766/sujst-2024-01-e02544.

[12] Sushma D S, Mohammed Alqhama, Aravind M, Jayanth A B, and Rakshith Kumar K, "Pest Detection and Classification in Peanut Crops," *International Research Journal on Advanced Engineering and Management (IRJAEM)*, vol. 2, no. 05, pp. 1372–1379, May 2024, doi: 10.47392/IRJAEM.2024.0189.

[13] "data mining and wireless sensor net work for groundnut pests thrips dynamics and predictions." Accessed: May 23, 2025. [Online]. Available: https://www.researchgate.net/publication/350175142_data_mining_and_wireless_sensor_net_work_for_groundnut_pests_thrips_dynamics_and_predictions

[14] M. Divya, T. N. Manjunath, and R. S. Hegadi, "A Study on Developing Analytical Model for Groundnut Pest Management Using Data Mining Techniques," in *2014 International Conference on Computational Intelligence and Communication Networks*, IEEE, Nov. 2014, pp. 691–696. doi: 10.1109/CICN.2014.152.

[15] "CEUR-WS.org/Vol-2831 - Workshop on Scientific Document Understanding 2021." Accessed: May 23, 2025. [Online]. Available: https://ceur-ws.org/Vol-2831/

[16] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2022, pp. 11966–11976. doi: 10.1109/CVPR52688.2022.01167.

[17] "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." Accessed: May 23, 2025. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[18] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10691–10700, May 2019, Accessed: Apr. 25, 2025. [Online]. Available: https://arxiv.org/pdf/1905.11946

[19] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2022, pp. 11966–11976. doi: 10.1109/CVPR52688.2022.01167.

[20] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2022, pp. 11966–11976. doi: 10.1109/CVPR52688.2022.01167.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[22] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf Process Manag*, vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/j.ipm.2009.03.002.

Utilizing Pre-Trained Models for Disease Detection in Peanut Cultivars through Deep Learning-Based Health Monitoring

Kashif Mahmood[1], Ume Habiba[1], Muhammad Ramzan[1]
[1]Department of Software Engineering, University of Sargodha, Sargodha, Punjab, Pakistan

*Corresponding author: Dr. Muhammad Ramzan (muhammadramzan@gmail.com)

Abstract:

Agriculture is a highly dynamic area that sustains global food security, and crop health plays a critical role in obtaining agricultural output. Peanuts, commonly known as groundnuts, hold a significant value due to their nutritional importance and economic benefits in many regions. However, it faces numerous challenges due to its vulnerability to a range of diseases that have severe impact on both quality and yield. Disease detection methods based on traditional techniques were time consuming, vulnerable to human mistakes and Labor intensive. To handle these issues, we suggest a cutting-edge approach for the detection of peanut diseases based on deep learning models. In this research work, we proposed some pre-trained deep learning models such as EfficientNet-B0, EfficientNet-B4, ConvNeXt-Base, and ConvNeXt-Small model. Deep learning models were trained in order to identify peanut leaf diseases using artificial intelligence for the purpose of training deep learning models with 1720 publicly available images of dataset. The dataset includes both healthy and diseased images of groundnut leaf such as Alternaria leaf spot, Rosette, Rust and Leaf spot (early and late). The dataset was split into 80% training images, 10% test images, and 10% value accuracy images. Out of all the proposed models, ConvNeXt-Small outperformed on the same dataset and gave the better results than the older-ones on same dataset. Earlier same dataset had the accuracy rate of 96.51%. Our experiments showed that the suggested model achieved a 98.91% accuracy rate, and outperformed the existing state-of-the-art models. Results on these benchmarks demonstrate the merit of the proposed model for application in real agricultural problem, providing a new benchmark for groundnuts leaf disease detection, and establish the feasibility of AI-powered solutions for enhancing transferable sustainable agricultural practices.

Introduction:

The word agriculture encloses a broad range of practices which domesticated animals and crop plants use to support the global need of human population's sustenance and other essential products. Agriculture's critical demanding function has been widely acknowledged for a long time in development. The speed at which technological advancements are happening in agricultural field have resulted in the improved output and efficiency, ultimately enhancing the economic expansion. AI powered solutions such as using technologies like machine learning, deep learning, IOT-based systems etc will enable the farmers to generate more output with less input(1).

Arachis Hypogaea commonly known as peanut is the highly acknowledged oilseed-based plant of nuts family. In context of enriching soil, livestock food, and human food consumption with nitrogen, peanut serves as an oil plant(2). Its kernels are rich in essential minerals and vitamins. Despite all these benefits of groundnut, farmers are facing problems in its cultivation due to several prevalent diseases. According to the estimation of the United Nation's Food and Agriculture Organization, annually between 20 and 40% of world's crop production is destroyed due to infestations and diseases, even though roughly two million tons of pesticides are being used(3).

Peanut growth is significantly disrupted by the leaf diseases which can affect the quality and yield in a negative manner. That's why peanut leaf diseases identification is of the utmost interest. The detection of groundnut diseases in time and accurately is of much importance.

Traditionally techniques that were being used to detect the peanut leaf diseases were manual. They required a lot of human effort and time. Due to the complex environmental conditions and poor distinction between peanut diseases symptoms, it becomes hard to detect groundnut diseases by using conventional methods and reduces the quality and efficacy of crop yield. Evaluating the early-stage condition of leaf diseases can be challenging primarily because of inadequate number of skilled personals and research tools(4).

In the recent times, there are several new techniques that are being used to address all these issues and identify the peanut diseases problems. Over the past few years, deep learning computational models have proven its increasingly crucial role in the diagnosis of crop leaf diseases and showed better performance than many other traditional methods(5).

In our research, we used several deep learning models to detect the peanut diseases. Deep learning techniques have garnered a lot of attention and consideration as a result of their promising and massive outputs in various applications such as classification and identification of groundnut leaf diseases. Several models such as EfficientNet-B0, EfficientNet-B4, ConvNeXt-Base, and ConvNeXt-Small are used to effectively evaluate the groundnut leaf diseases.

We used a dataset of total 1720 images including 5 classes as Alternaria Leaf spot, Healthy, Rust, Leaf spot (early and late) and Rossette.

In our paper, our main work is to identify and detect the groundnut diseases using some deep learning algorithms. Following are the few contributions of this study's proposed work:

- We used pre-trained deep learning models to get the enhanced and accurate results.
- We divided our each of the five classes of our dataset into training, testing, and value accuracy images.
- We used four different pre-trained models as EfficientNet-B0, EfficientNet-B4, ConvNeXt-Base, and ConvNeXt-Small to analyze the different deep learning architectures and comparing each of them to get the best output.
- ConvNeXt-Small model gave the best outcomes in our experimentation and showed better accuracy than the previous experimentation on the same publicly available dataset.
- Accuracy, Precision, F-measure, and Recall are measured by using the suggested models in our work.

The subsequent sections of this paper will be structured as follows: In section 2, we will be reviewing the relevant literature. Section 3 will provide detailed explanation of the models being utilized in the research work and their detailed methodology. Detailed description of the dataset being used in the paper will be given. Last section will be concluding the paper by providing all the results of the models being used.

Literature review of existing papers on Peanut diseases:

Peanut diseases are no different than other plant diseases that are being notably affected, and plant diseases are a great threat to any agricultural production. Numerous studies have been done in the field of groundnut disease detection. In this section, various studies conducted on detection of peanut disease are presented. As a result, the main existing papers in the most related work in this domain are mentioned below under the sections of deep learning and machine learning papers.

Existing papers on groundnut diseases that used deep learning models

Usually, deep learning is employed to use neural networks with more than one hidden layer to classify and find regions of interest in an image, for example plant disease symptoms. It builds an end-to-end system which processes leaf images and predicts the likelihood of leaf to be diseased with specific diseases.

Sasmal et al (6) designed a dataset containing total 1720 original groundnut images. The dataset included the peanut images as Healthy, Alternaria leaf spot, Rust, Leaf spot (early and late), and Rosette which was carefully obtained from the West Bengal, India. They utilized several models as InceptionV3, DenseNet201, AlexNet, and ResNet50 and gained the highest accuracy of 96.51%. Urbano et al (7) used deep learning techniques such as VGG16, VGG19, MobileNet, DenseNet, Xception, InceptionResNetV2, InceptionV3, and ResNet50 to identify peanut leaf spot diseases. Researchers used the dataset including 1000 images to identify leaves having leaf spot diseases. They achieved the highest accuracy of 98%. Anbumozhi et al (8) primarily focused on the identification of self-collected dataset under several environmental conditions. They proposed the Progressive Groundnut Convolutional Neural Network (PGCNN) model for the disclosure of leaf diseases at the early stages. They gained the overall accuracy of 97.58%. Paramanandham et al (9) suggested the LeafNet model to identify and detect the peanut leaf diseases. Their dataset included the total 10,361 images including the 6 classes of diseased as well as healthy images. The testing accuracy obtained by them was 97.225%. Aishwarya et al (10) performed their work on self-created dataset. They used the Tri-CNN ensemble architecture models such as DenseNet169, Inception, and Xception models. Their proposed model performed well on the real-world peanut leaf dataset and achieved the accuracy rates of 98.46%. Rakholia et al (11) proposed a deep learning model in which progressive resizing was effectively performed to identify peanut leaf disease and for the classification work. They involved the five categories of peanut leaf diseases dataset. In their paper, accuracy obtained was 96.12%. Chen et al (12) suggested two methods of imaging such as push-boom FX10 and Snapshot. Peanut issues were detected by using the deep learning models. Their dataset included total 1560 images and attained the overall maximum accuracy of 98.5%. Wang et al (13) used the faster RCNN method. In their work, dataset including 1300 sets was collected to perform the experimentations. Lv et al (14) collected the dataset consisting of 700 images of peanut rust diseases, scorch spot, and leaf spot for the identification of peanut leaf diseases. They employed YOLOX-Tiny network for the experimentation purposes of groundnut leaf diseases and achieved the enhanced accuracy. Sivaganesan et al (15) proposed various CNN models as VGG-16, EfficientNet, and MobileNet for the identification of the leaf diseases. Rajmohan et al (16) used the peanut images dataset to detect the peanut leaf diseases at the early stages. They employed the convolutional neural network and k-means clustering methodology to correctly detect the infected images. They gained the accuracy of 93%.

Table 1 is used to represent the literature review of existing papers of Groundnut diseases based on deep learning models.

Table 1: Existing papers of peanut diseases based on deep learning techniques

Reference	Year	Dataset	Models
(6)	2024	1720 images	Deep learning models
(7)	2022	1000 images	VGG16, Xception, VGG19, MobileNet, InceptionV3, DenseNet, ResNet50, and InceptionResNetV2
(8)	2023	Self-collected	Progressive Groundnut Convolutional Neural Network (PGCNN), VGG Models, AlexNet, CNN
(9)	2024	10,361 images	LeafNet
(10)	2023	Self-collected	tri-CNN ensemble model as DenseNet169, Inception, and Xception models

(11)	2022	Manually created	Deep learning model
(12)	2023	1560 peanuts images	Deep learning
(13)	2023	1300 sets	Faster RCNN
(14)	2025	700 images	YOLOX-Tiny network
(15)	2023	Groundnut images	different CNN models, such as MobileNet, EfficientNet, and VGG-16
(16)	2022	Peanut leaf images	CNN model and k-means method

Existing papers on groundnut diseases that used machine learning models

Gowrishankar et al (17) proposed the machine learning models support vector machine (SVM), and such as artificial neural network (ANN) for the examination of peanut leaf diseases. They used the groundnut images and integrated several image processing methods for detection purposes. In their experiments, they found out that ANN performed better than SVM. Zou et al (18) utilized a dataset of total 600 images including the five kinds of groundnut for the identification of peanuts mildew. They used several machine learning models as GBDT (Gradient Boosting Decision Trees), XGBoost (Extreme Gradient Boosting), LightGBM (Light Gradient Boosting Machine), and CatBoost in their work. At the end, LightGBM proved to be the best among all showing the identification rate of 99.10%. Existing papers of Peanut diseases based on machine learning models are shown in table 2.

Table 2: Existing papers of peanut diseases based on machine learning techniques

Reference	Year	Dataset	Models
(17)	2020	Groundnut Images	Machine learning→ Artificial Neural Network (ANN), Support Vector Machine (SVM)
(18)	2022	600	Several Machine learning models as XGBoost (Extreme Gradient Boosting), GBDT (Gradient Boosting Decision Trees), LightGBM (Light Gradient Boosting Machine), CatBoost

Proposed Research Methodology:

The knowledge of the system under study is essential for the formulation of the methodology for the research, and this is why the proposed research methodology section is a vital tool in sketching the systematic approach that will be adopted in order to attain the study's objectives. It provides a detailed blueprint for the procedures, tools and techniques that will be used to gather and analyze data to entice with research that has been conducted in a rigorous and transparent manner. The complete description of data, suggested models and their detailed overview, and performance measures are included in this section in order to secure the credibility and validity of the research. Fig 1. is representing the overall methodology of our proposed models used in our research work.

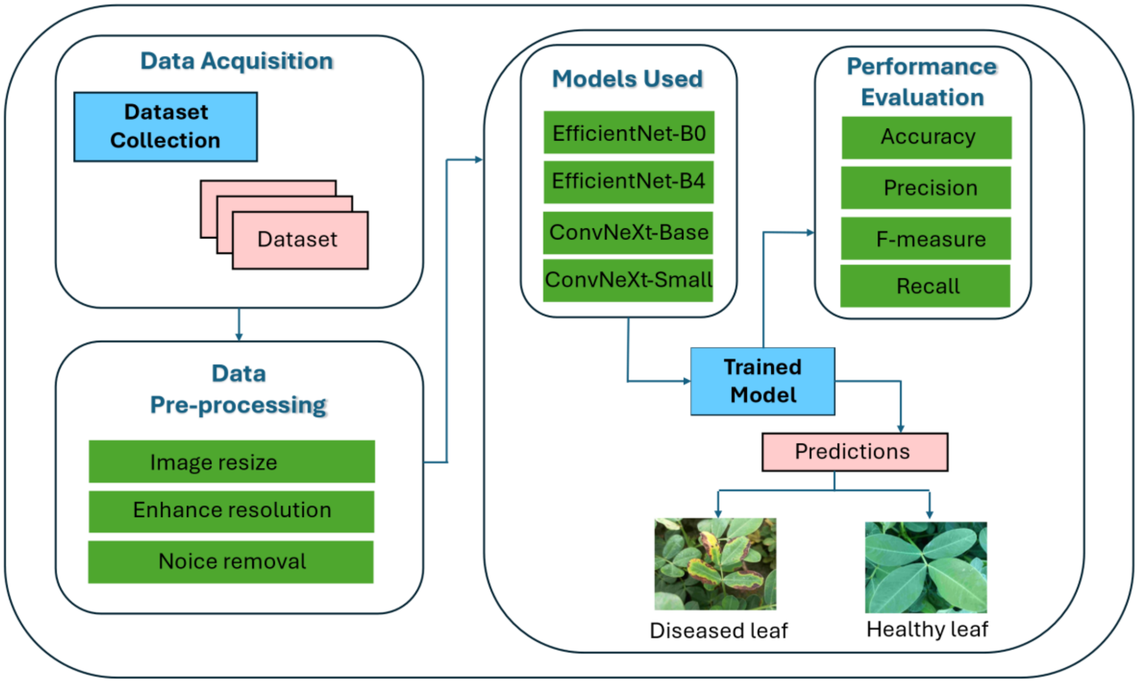


Fig 1. Proposed architecture workflow of Peanut diseases detection

Data acquisition for our work

A publicly available dataset (6)was used for the research purposes. In original dataset, groundnut leaves that were diseased with Leaf spot (early and late), Rust, Alternaria leaf spot, Rosette, and Healthy had been collected from agricultural land and brought for analysis. The dataset holds 1,720 JPG files, all in the pixel format together with the associated image number and names. The healthy data, Leaf spot (early and late) images, Alternaria leaf spot images, Rust and Rosette diseases images data was respectively uploaded to 5 different folders; healthy data, Leaf spot (early and late), Rossette, Alternaria leaf spot, and Rust. Each folder contained further 3 folders indicating as test, train, and value accuracy images for each disease. Images were divided as 80% training, 10% testing, and 10% value accuracy images. Fig 2. include the five classes of peanut images present in our dataset.

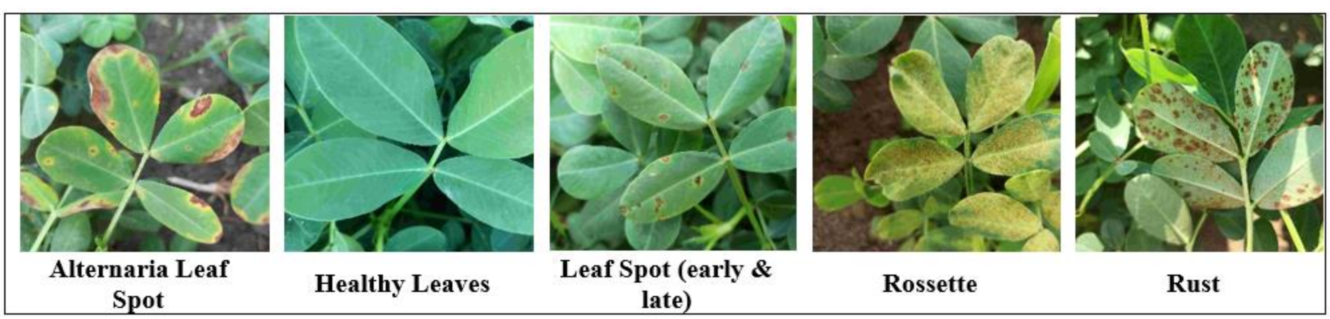


Fig 2. Showing the images included in our Dataset

Additionally, the name of each folder was the same as the corresponding image class. The Healthy class folder contains a group of images of groundnut leaf in a healthy state. Photos of groundnut leaves affected with leaf spot (early and late) are given in the Leaf Spot (early and late) folder. In the folder titled Alternaria Leaf Spot are images of groundnut leaves which are infected by Alternaria leaf spot. Images of groundnut leaf infected with Rust disease are contained in the Rust named folder. The Rosette folder holds the images of peanut leaves affected by Rosette disease. DISTINCT directories were set up for the experimental procedures on images data for loading the data. The most frequently affected diseases are selected for the procedure of leaf disease identification. Complete details of the dataset used in our research work is demonstrated in table 3.

Table 3: Description of used dataset in detail

Class name	Number of images	Images in train folder	Images in test folder	Images in value accuracy folder
Healthy	600	480	60	60
Leaf spot (Early and Late)	450	360	45	45
Alternaria Leaf Spot	450	360	45	45
Rust	120	96	12	12
Rossette	100	80	10	10
Total	1720	1376	172	172

In all these classes different number of images are present. We carefully curated all these images dataset in the separate folders to avoid any further issues. A visual overview of the dataset in shown below in the fig 3.

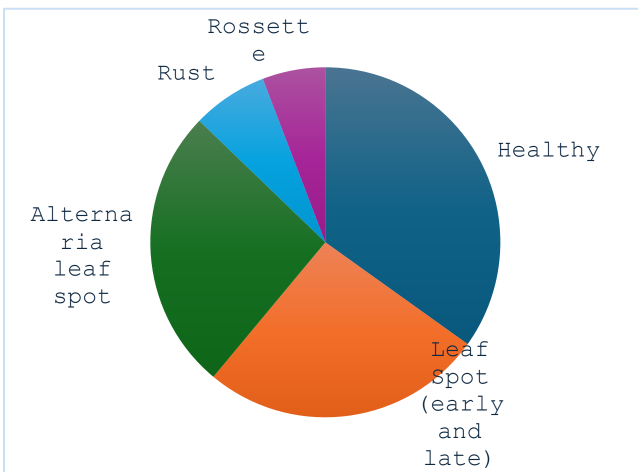


Fig 3. Visual representation of dataset

Pre-processing of Images

It is very important to mention that image preprocessing in DL is very important, it does increase model efficiency and accuracy of the model. The dataset taken for research was adopted from a publicly available source that had already done necessary preprocessing steps on it. The dimensions of the images were also made uniform by first processing the images to give uniform dimension. The non square images were cropped into square dimensions including the full leaf area. To better analyze further, all images were resized to 224 × 224 pixels. Besides, augmentation techniques on images, including rotation, scaling, flipping, cropping, and adding noise, were considered to improve the diversity and quantity of the training data; however, in this study these techniques were applied because the dataset to test the performance of a deep learning model would suffice. Instead, with this approach we were able to begin training and evaluation of models without any additional preprocessing. Images were given to the pre-trained deep learning model after the application of preprocessing techniques.

Feature extraction and classification

In our research work, we proposed and applied five deep learning pre-trained models as EfficientNet-B0, EfficientNet-B4, ConvNeXt-Base, and ConvNeXt-Small:

EfficientNet-B0 model structure:

Mingxing Tan and Quoc V. Le, authors of the paper "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," proposed EfficientNet-B0(19)

In that respect, EfficientNet is a completely different way of exploring the limits of convolutional neural networks (CNNs): models only scaled in either depth, or in resolution or width, none combined at once. However, there was no systematic methodology in this approach. In order to solve this problem, Google Brain researchers implemented the EfficientNet model which used a composite scaling method. It enhances all three dimensions in proportion to obtain the optimal results by using a set of scaling coefficients which are defined. The EfficientNet-B0 is the smallest of the family and has less parameters and lower computational cost that enables it to perform competitively on resource constrained devices with relatively high accuracy.

The EfficientNet-B0 consists of one Stem layer, multiple Inverted Residual Bottleneck (IRB) blocks, one Head layer and one fully connected layer. Specifically:

- **Stem Layer**: This image has been convolved with a 3x3 convolution and a stride of 2 and a padding of 1 to output 32 channels. The initial feature extraction is taken care of by this layer.
- **Inverted Residual Bottleneck (IRB) blocks:** 8 IRB blocks total are used, each having a depth wise separable convolution, expansion layer, 3x3 deper convolution and squeeze & extication (SE) module. On the other hand, each block has varying number of channels, expansion ratio, and stride. The first IRB block is an example of 16 channels, expansion ratio of 1, stride of 1 and containing an SE module. Table 3 provides the detailed structure of the IRB blocks.
- **Head Layer:** Output of the IRB blocks: After 1 × 1 convolution with 1280 channels and 7 × 7 adaptive average pooling layers to down sample the feature map size, the outcomes of the IRB blocks will pass through this stage.
- **Fully Connected Layer:** The output of the Head layer is flattened and fed to a fully connected layer with 1000 neuron (that can be a number of classes for a particular task), and the outputs generate classification results as final result.

Table 4: Layered Structure of EfficientNet-B0 model

Stage	Operator	Resolution	#Channels	#Layers
1	Conv3x3, s=2	224 × 224 → 112 × 112	32	1
2	MBConv1, k3x3, s=1	112 × 112	16	1
3	MBConv6, k3x3, s=2	112 × 112 → 56 × 56	24	2
4	MBConv6, k5x5, s=2	56 × 56 → 28 × 28	40	2

5	MBConv6, k3x3, s=2	28 × 28 → 14 × 14	80	3
6	MBConv6, k5x5, s=1	14 × 14	112	3
7	MBConv6, k5x5, s=2	14 × 14 → 7 × 7	192	4
8	MBConv6, k3x3, s=1	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7 → 1 × 1 → 1 × 1	1280 → 1000	1

The key to EfficientNet is its compound scaling method, which simultaneously scales the width, depth, and resolution of the network using fixed scaling coefficients. The formula is:

$$\phi=\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \quad (1)$$

The scaling coefficients for depth, width and resolution are designated here to be α, β and γ, respectively. To compute $\alpha = 1.2$, $\beta = 1.1$, and $\gamma = 1.15$ on the baseline network EfficientNet-B0 ($\Phi = 1$), the authors performed a grid search. Thus, with these coefficients fixed, different Φ values are used to scale EfficientNet-B0 in to form a family of neural networks: EfficientNet-B1 to B7.

Mobile inverted bottleneck MBConv blocks with squeeze and excitation optimization are used in baseline network, mathematically expressed as:

$$MBConv(x)=SE(DepthwiseConv(ExpandConv(x))) + x \quad (2)$$

where the expansion convolution increases channel dimensionality before depthwise spatial filtering, followed by channel-wise feature recalibration through squeeze-excitation (SE) layers. EfficientNet-B0's architectural efficiency stems from its optimal $\phi=1$ configuration, achieving 77.1% ImageNet accuracy with only 5.3M parameters through this balanced scaling approach.

EfficientNet-B4 model structure:

EfficientNet-B4 was first proposed by Mingxing Tan and Quoc V. Le in the paper "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks"(19)

Prior to EfficientNet, known scaling of CNNs mostly came via increasing depth, width, or resolution individually. However, this process had no systematic methodology. In order to address this issue, Google Brain researchers introduced EfficientNet model family based upon a compound scaling method. It scales in all three dimensions proportionally using defined scaling coefficients in order to get optimal performance without separately increasing depth, width or resolution. The EfficientNet family has medium sized model named as EfficientNet B4 which is an adjustment in between the accuracy and computation efficacy. It was trained using a multi-objective neural architecture search with the same aim as MnasNet, of optimizing accuracy as well as FLOPs.

Specifically, EfficientNet-B4 contains a Stem layer, multiple Inverted Residual Bottleneck (IRB) blocks, a Head layer, and a fully connected layer.

Specifically, the architecture is as follows:

- **Stem Layer:** 3×3 convolution with stride of 2, padding of 1 on input image to get 48 channels which is output from Stem Layer. It is this layer, which is the first to take care of the initial feature extraction.
- **IRB Layer:** The total number of IRB blocks is 14, each containing a (depthwise separable) convolution layer, an expansion layer, a (depthwise) 3 × 3 convolution layer, and a squeeze-and-excitation (SE) block. Across blocks, number of channels, expansion ratio and stride are different. For instance, the first IRB block has 24 channels, an expansion ratio of 1, a stride of 1 and has an SE module.

- **Head Layer:** output of these blocks is then fed to 1×1 convolution of 1792 channels and 7×7 adaptive average layer of pooling to remove the spatial dimension of maps feature.
- **Fully Connected Layer:** This means that the output of the Head layer is flattened and passed to a Fully Connected layer with 1000 neurons for a final classification since we will be using 1000 neurons as per the task.

Table 5: layered structure of efficientnet-b4 model

Stage	Operator	Resolution	#Channels	#Layers
1	Conv3x3, s=2	380 × 380 → 190 × 190	48	1
2	MBConv1, k3x3, s=1	190 × 190	24	1
3	MBConv6, k3x3, s=2	190 × 190 → 95 × 95	32	2
4	MBConv6, k5x5, s=2	95 × 95, 48 × 48	56	2
5	MBConv6, k3x3, s=2	48 × 48 → 24 × 24	112	3
6	MBConv6, k5x5, s=1	24 × 24	160	3
7	MBConv6, k5x5, s=2	24 × 24 → 12 × 12	272	4
8	MBConv6, k3x3, s=1	12 × 12	448	1
9	Conv1x1 & Pooling & FC	12 × 12 → 1 × 1 → 1 × 1	1792 → 1000	1

Its compound scaling method comprises scaling in network (width, depth, and resolution) simultaneously with defined coefficients of scaling being the core of EfficientNet.
The formula is:

$$\phi = \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \tag{3}$$

Here, α, β, and γ are the scaling coefficients for depth, width, and resolution, respectively. Based on the baseline network EfficientNet-B0 (Φ=1), the authors determined α=1.2, β=1.1, and γ=1.15 through grid search. With these coefficients fixed, different Φ values are used to scale EfficientNet-B0 to derive a family of neural networks, EfficientNet-B1 to B7. EfficientNet-B4 corresponds to Φ=4.
The total FLOPS increase by a factor of $(\alpha \cdot \beta 2 \cdot \gamma 2)\phi$ when scaling the network.

EfficientNet-B0 model's architecture contain seven blocks and the most important building block of this model is MBConv, which is a mobile inverted bottleneck convolution(19). EfficientNet-B4 model scaled up from the EfficientNet-B0 model by using the method of compound of compound scaling. This model is having more depth and its blocks are very deep [19].

Below figure 3 is showing the basic layered framework of the EfficientNet model.

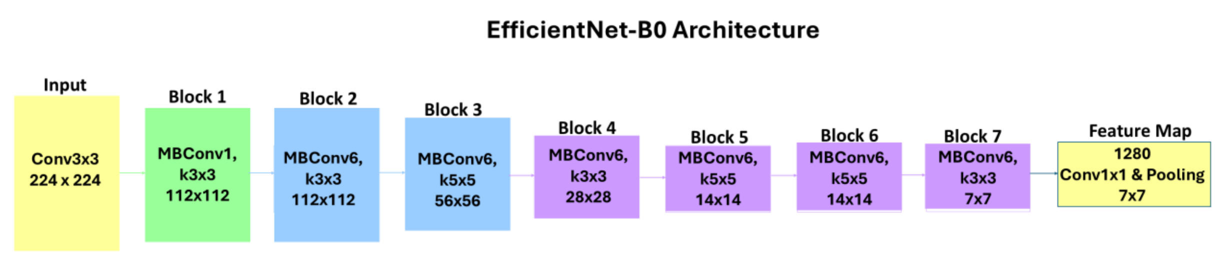


Fig 4. Layered architecture of EfficientNet-B0 model

ConvNeXt-Base model structure:

The ConvNeXt model was first proposed by Liu et al. in the paper "A ConvNet for the 2020s"(20) published in 2022. Convolutional neural networks (CNNs) have been profitably applied to the functions of classifying images over the recent years, due to the improvements and advancements of deep learning. But in the last recent years, ViTs have also shown good performance in many vision tasks. Based on ViTs, the authors of ConvNeXt reinvented the design space of CNNs and proposed the ConvNeXt model. Inherit core ideas of CNNs and some design ideas from ViTs, like hierarchical feature maps and self attention mechanisms, ConvNeXt is the model. This achieves competitive performance with the advantages of CNNs in terms of computational efficiency and interoperability. Four stages with each stage including multiple blocks are the components of ConvNeXt-Base.

The full architecture is as follows:

- **Stage 1:** 4×4 convolution of the input image with stride 4 thus resulting in 96 channels output. It is the first feature extraction stage.
- **Stage 2:** This stage contains 3 blocks. The block is each 7×7 depthwise convolution followed with a LayerNorm module and 1×1 pointwise convolution and another 1×1 pointwise convolution with GELU. There are 192 channels in this stage.
- **Stage 3:** 9 blocks are present in this step, and the structure is same as of the blocks in the Stage 2. The number of channels in this stage is 768.
- **Stage 4:** The output of this stage is the full classification results.

The model employs a 4-stage hierarchical structure with progressive downsampling through strided convolutions:

$$Stage_i(x) = ConvNeXtBlock^{N_i}(Patchify(x)) \tag{4}$$

where N1=3*N*1=3, N2=3*N*2=3, N3=27*N*3=27, N4=3*N*4=3 blocks per stage, and patchify layers use 4×4 kernels for spatial reduction.
Each ConvNeXt Block implements:

$$x_{out} = x + DW(Conv(GELU(LayerNorm(MLP(x))))) \tag{5}$$

utilizing 7×7 depthwise convolutions for large receptive fields and inverted bottlenecks with 4× expansion ratios. The base configuration contains 88M parameters, achieving 84.1% ImageNet accuracy through its macro design that mimics Swin Transformers' stage compute distribution.

ConvNeXt-Small model structure:

However, as deep learning has come a long way, convolutional neural networks (CNNs) have already proven their great prowess in image classification tasks. In recent years there are several vision tasks for which ViTs outperform by strong margins. The authors of ConvNeXt re-explored the design space of CNNs based on ViTs and proposed the ConvNeXt model. ConvNeXt model puts the main thoughts of CNN in its mind, only with some design ideas of ViTs such as hierarchical feature maps and self attention mechanisms. In the meantime, it strikes a great harmony between CNN advantages (computational efficiency, interoperability) and competitive achievement. Each stage in ConvNeXt-Small contains several blocks, and they are four in total.

The detailed architecture then is as such:

- **Stage 1:** The input image is 4x 4 convolved with a stride of 4 producing 96 channels. It is first feature extraction stage.

- **Stage 2:** This stage contains 3 blocks. For each block, we have a 7×7 depthwise conv, a LayerNorm module, a 1×1 pointwise conv and a 1×1 pointwise conv with a GELU activation. This stage has 192 channels.
- **Stage 3:** This stage contains 3 blocks, with the same structure as the blocks in Stage 2. The number of channels in this stage is 384.
- **Stage 4:** This stage contains 9 blocks, with the same structure as the blocks in Stage 2. There are total 768 numbering of channels present in this stage.
- **Classification Head:** The output of Stage 4 is processed by a global average pooling layer, followed by a fully connected layer to produce the final classification results.

The top-performing ConvNeXt-Small variant incorporates V2 enhancements including Global Response Normalization (GRN).
The modified block structure becomes:

$$x_{out} = x + \gamma(GRN\left(DWConv\left(GELU\left(LN(MLP(x))\right)\right)\right)) \qquad (6)$$

where $\gamma\gamma$ is a learnable scaling parameter and GRN implements cross-channel feature competition:

$$GRN(x) = \frac{x}{\sqrt{\frac{1}{C}\sum_{C=1}^{C} x_C^2 + \in}} \qquad (7)$$

This normalization enhances feature discriminability for subtle disease patterns. With 50M parameters and 15.4B FLOPS, the small variant achieves 83.6% ImageNet accuracy through optimized stage allocation (N1=3N1=3, N2=3N2=3, N3=9N3=9, N4=3N4=3). The architecture's success in almond disease detection stems from its combination of large-kernel spatial processing and GRN-enhanced feature learning, particularly effective for small lesion localization in high-resolution orchard imagery.
ConvNeXt model is primarily a highly connected deep learning model which consists of convolutional neural networks(21).
Fig 5. Is depicting the basic layered structure of ConvNeXt model.

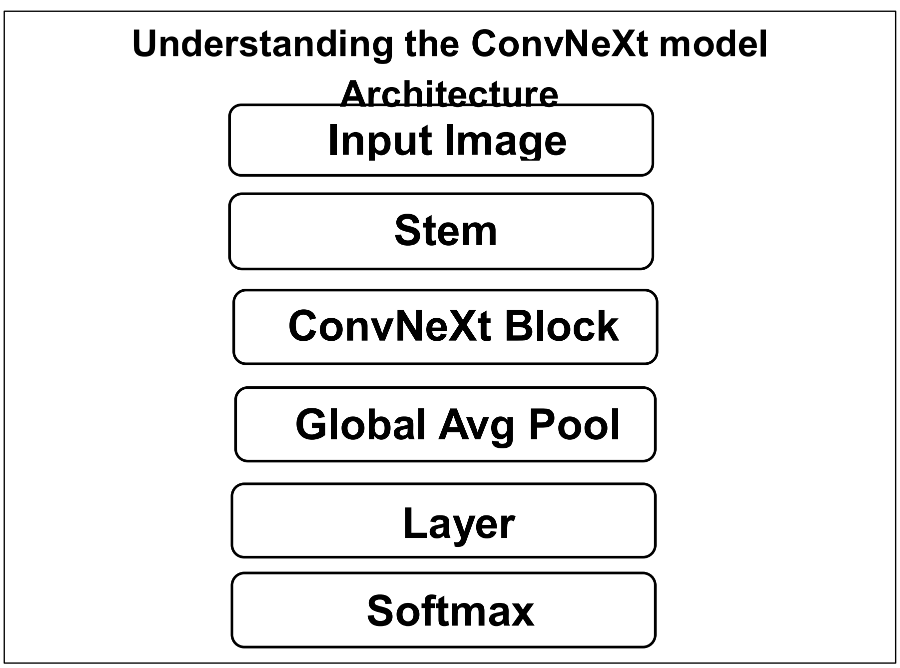


Fig 5. General structure of ConvNeXt model

Performance measures

Conventional methods for estimating deep learning classifiers rely on confusion metrics that compare the rock-bottom ground truth in the dataset with the model's predictions, specifically using TP, TN, FP, and FN to represent true positives, true negatives, false positives, and false negatives, respectively.

Accuracy:

Accuracy of a model is how accurate it is at spotting the outcome. The number of correct outcomes can be divided by the total predicted to get accuracy.
The fraction proportion of the outcomes that are successfully categorised as follows expressed using the equation:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{8}$$

Precision:

Precision is measured in a way that true positive is the number of instances, for which predicted value by the model is positive, and it is correct; and total positive instances, which includes wrong and correct predicted positive values. This can be seen as follows:

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

F-measure / F-score:

F-measure (also known as F1 score) balances Precision and Recall by taking their harmonic mean. It provides a single metric to evaluate the model when both Precision and Recall are important and there is an uneven class distribution.

$$F = \frac{2X\ precision \times recall}{precision+recall} \tag{10}$$

Recall:

Recall or sensitivity (also known as the actual positive rate) is used to measure the model accuracy at identifying actual positive instances.

$$R = \frac{TP}{TP+FN} \tag{11}$$

Results and discussion:

In the section below, a detailed analysis of the results is provided which is obtained by using these four models: EfficientNet-B0, EfficientNet-B4, ConvNeXt-Base, and ConvNeXt-Small. Each model is being evaluated across various parameters as accuracy, precision, recall, and F-measure.

Results obtained by using EfficientNet-B0 model

With an accuracy of 87.50%, EfficientNet-B0 is azure by an overall precision of 92.13%, but lacks recall at 78.88% and F1-score of 81.21%.

Classification report metrics of EfficientNet-B0 model:

The performance across different classes is quite variable in a classification report of EfficientNet-B0, shown in fig 6. The model has a very high precision (92.13%) however its recall (78.88%) is extremely low, which is 81.21% according to F1 score. More specifically, the 'Healthy' class has a remarkable precision (95.6%) as well as recall (98.3%) of healthy samples, which shows strong performance on healthy samples. Despite such problems, the recall of the ‘Leaf Spot (Early and Late)’ class (17.5%) is poor and indicates that early-stage leaf spot conditions cannot be easily identified. Recall is 37.5% and precision is 30.0%, due at least in part to the visual similarity to other stress induced patterns of imagery that belong to the 'Rosette' class.

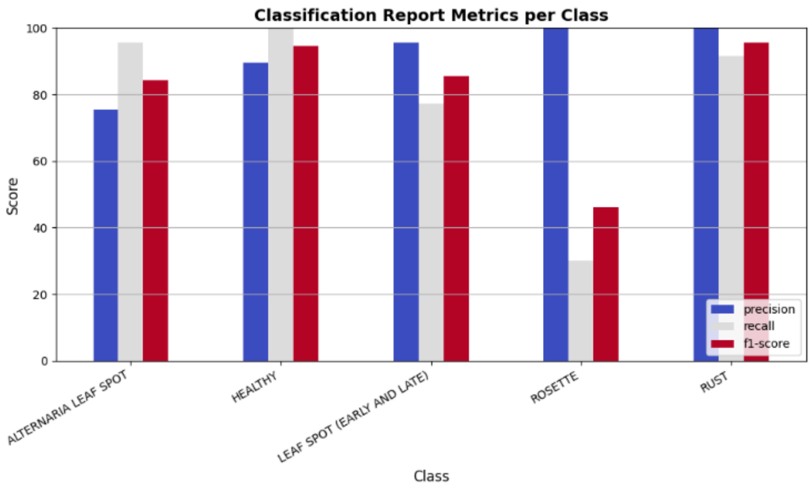


Fig 6. Classification report using EfficientNet-B0 model

Confusion matrix using EfficientNet-B0 model:

Fig 7. further shows these challenges in the confusion matrix. Since the model classifies correctly for most samples, and the diagonal dominance shows it, the model mistakenly classifies "Rosette" with "Alternaria Leaf Spot" (6 misclassifications), and "Healthy" with "Leaf Spot (Early and Late)" (also 6 misclassifications). The lower recall in problematic classes is impact by these errors.

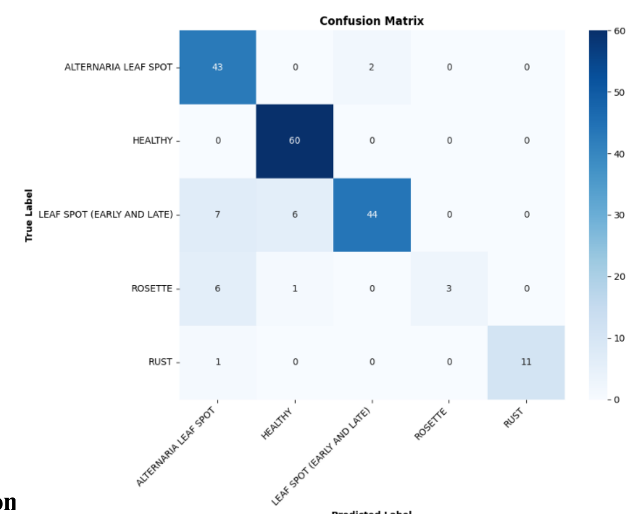


Fig 7. confusion

Loss dynamics related to EfficientNet-B0 model:

As shown in fig 8(a), there is a steady decline of training loss up to the 20th epoch, and it converges at around 0.85. The fact it is effective learning during training indicates it is effective learning. But, although the validation loss (not shown) is slightly up after Epoch 15, it indicates a small amount of overfitting.

Training VS Value accuracy using EfficientNet-B0 model:

The accuracy curves in fig 8(b) shows 92.13% training accuracy and 87.50% validation accuracy. This will confirm that overfitting, since the model memorizes training patterns and generalizes bad on validation data.

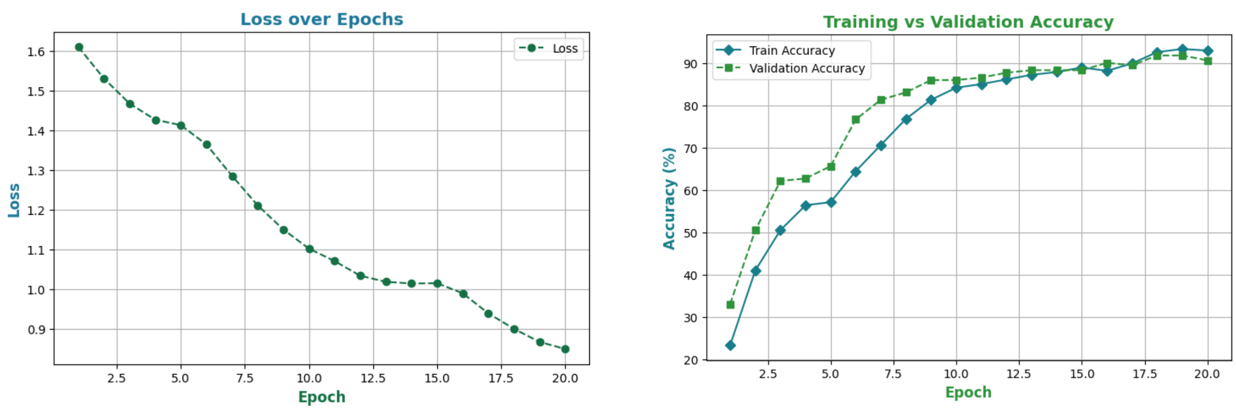


Fig 8. Loss dynamics using efficientnet-b0 model (a) Showing the loss dynamics (b) Showing the training vs validation accuracy

Results obtained by using EfficientNet-B4 model

EfficientNet-B4 model represented the 91.85% accuracy, precision of 74.61%, recall as 78.06%, and F1-score of 76.27%.

Classification report metrics of EfficientNet-B4 model:

The performance of different classes is shown in the fig 9. It can be observed in the figure that rust class showed the exceptional performance on the metrics of precision, recall, as well as f1-score.

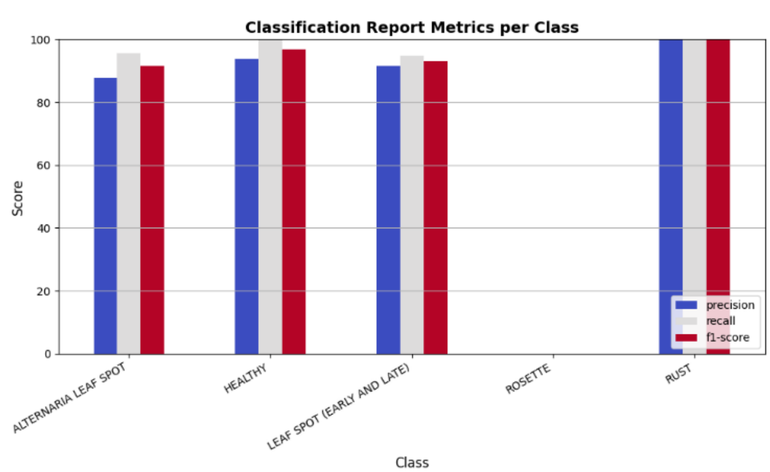


Fig 9. Classification report using EfficientNet-B4 model

Confusion matrix using EfficientNet-B4 model:

Fig 10. illustrate the confusion matrix of all the classes using the EfficientNet-B4 model. This confusion matrix will provide the deeper analysis of the proposed model on the dataset.

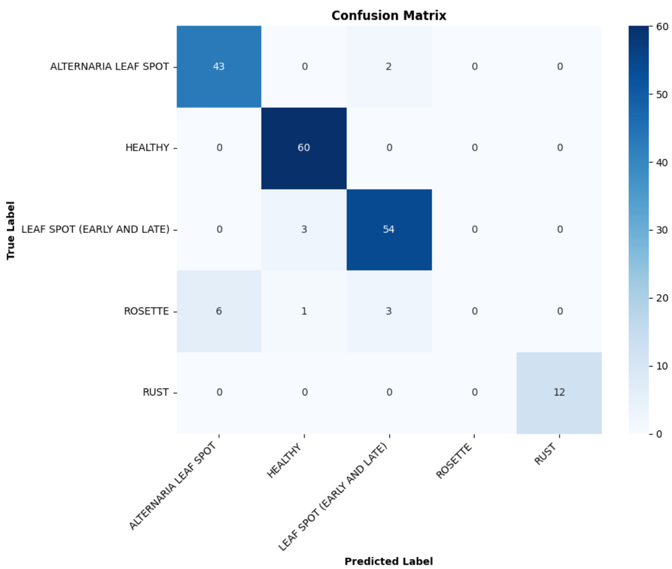


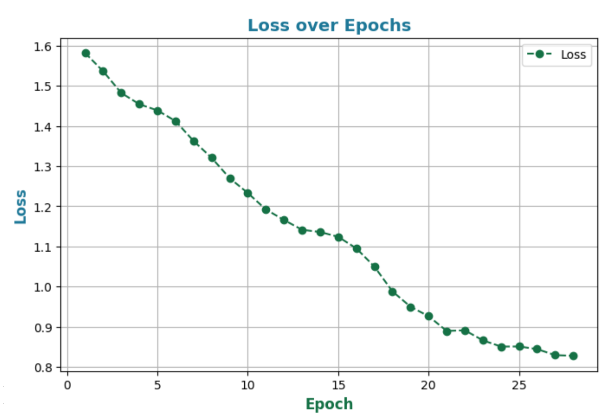
Fig 10. Confusion matrix using EfficientNet-B4 model

Loss dynamics related to EfficientNet-B4 model:

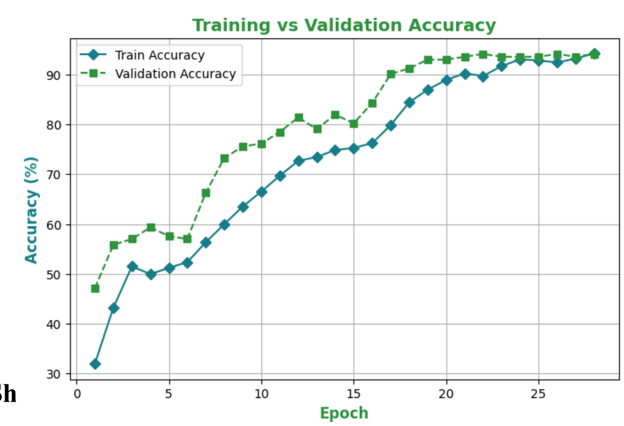
In fig 11(a), it can be seen that there is a stabilization over the 0.95. These loss metrics help to understand more about the used model in the research paper.

Training VS Value accuracy using EfficientNet-B4 model:

The training accuracy of 94.7% versus the 91.85% validation accuracy is shown in the fig 11(b). There is an overfitting confirmation as the gap is about 2.85% and that can be due to the deeper structure of model's architecture.



Sh



validation accuracy

Results obtained by using ConvNeXt-Base model

Accuracy shown by ConvNeXt-Base model is 96.20%. The values of other parameters such as precision, recall, and f1-score are 97.61%, 87.65%, and 90.02% respectively.

Classification report metrics of ConvNeXt-Base model:

The classification report clearly shows that various classes performed well using the ConvNeXt-Base model. Fig 12. demonstrates the results of all the classes as Healthy, Alternaria leaf spot, Leaf spot (early and late), rust, and rosette using the ConvNeXt-Base.

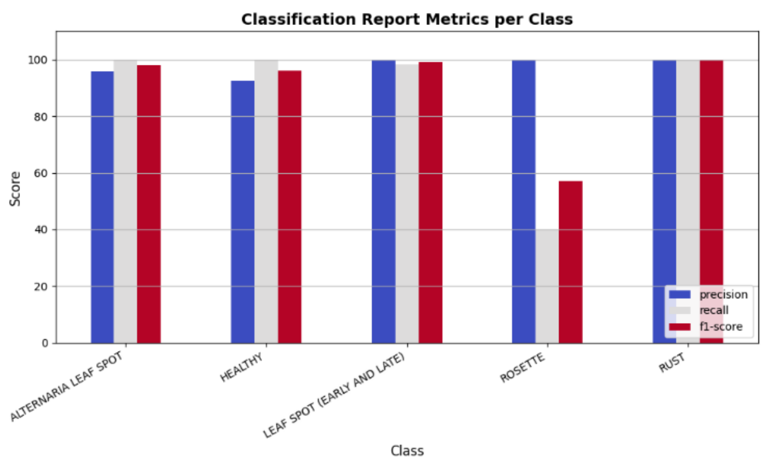


Fig 12. Classification report using ConvNeXt-Base model

Confusion matrix using ConvNeXt-Base model:

Visual representation of all the classes present in our dataset is shown in the figure 13 which is representing the confusion matrix of all the peanut image classes using the ConvNeXt-Base model.

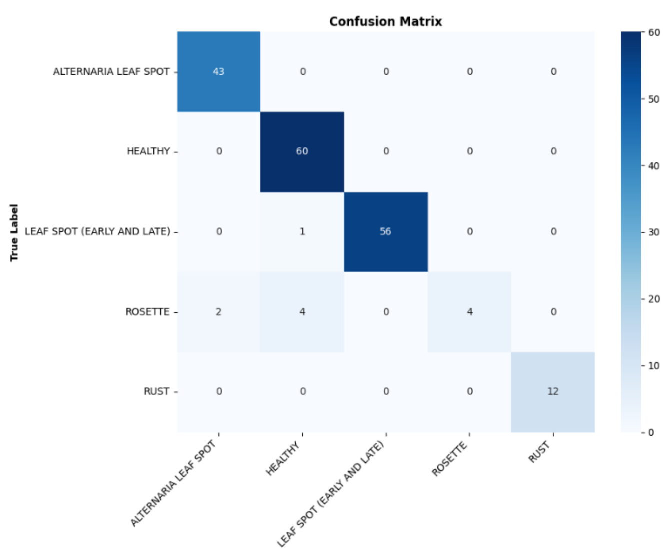


Fig 13. Co

Loss dynamics related to ConvNeXt-Base model:

Loss dynamics are shown in the fig 14. (a) which appeared by using the ConvNeXt-Base model in our work.

Training VS Value accuracy using EfficientNet-B4 model:

By using the ConvNeXt-Base model in our work, the validation accuracy of 96.20% was obtained. Fig 14. (b) represents a visual view of training versus validation accuracy.

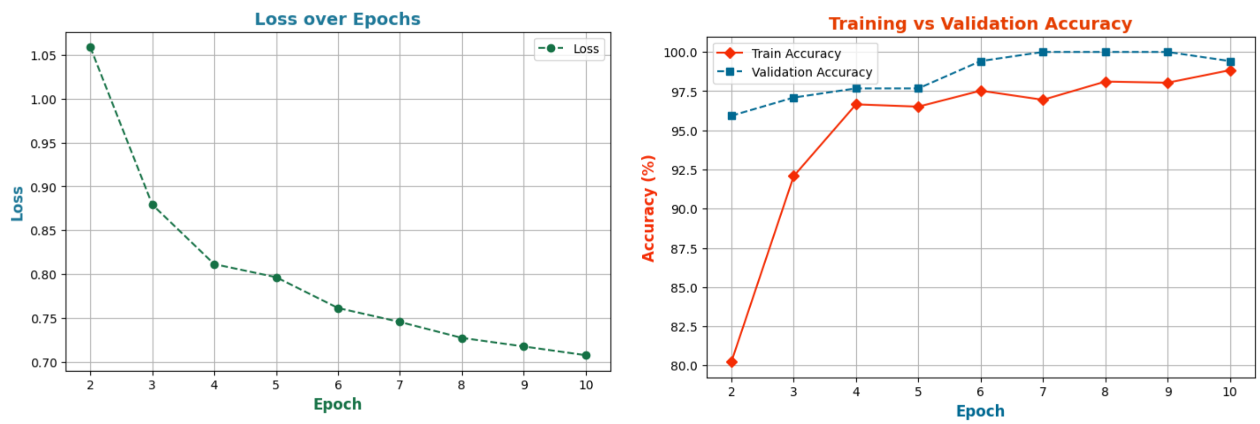


Fig 14. Using ConvNeXt-Base model (a) Showing the loss dynamics (b) Showing the training vs validation accuracy

Results obtained by using ConvNeXt-Small model

ConvNeXt-Small model showed the exceptional results in our research work and proved to be the best among other modernized models applied on the same publicly available dataset.
The highest accuracy shown in our work is 98.91% by using ConvNeXt-Small model. All other performance measure parameters showed the extraordinary results. Our experimental work got the Precision value of 99.15%, Recall of 99.30%, and F1-score as 99.21%.

Classification report metrics of ConvNeXt-Small model:

It is very evident from the fig 15. showing the classification report metrics of all the image classes of our dataset using the ConvNeXt-Small model that our model performed best on all the classes.

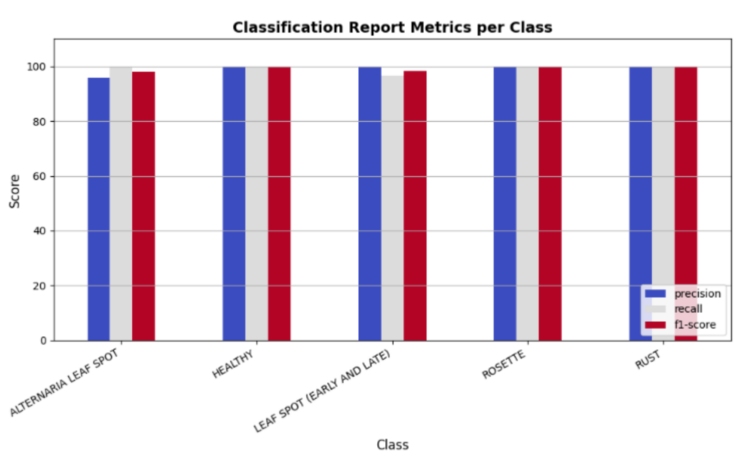


Fig 15. Classification report using ConvNeXt-Small model

Confusion matrix using ConvNeXt-Base model:

The confusion matrix made by using the ConvNeXt-Small model on the peanut images of our dataset is given in the fig 16.

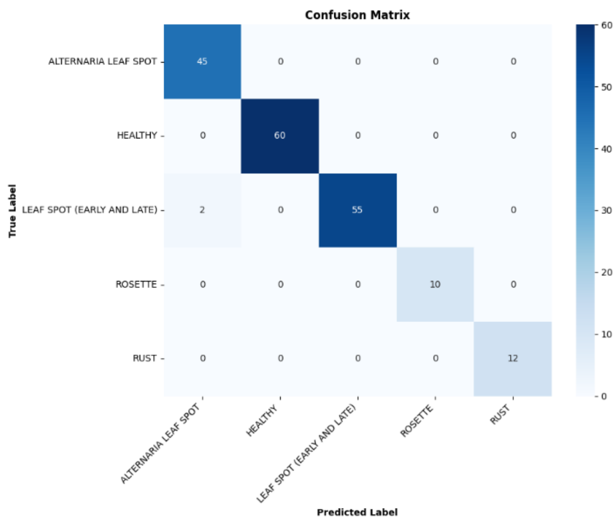


Fig 16. Confusion matrix using ConvNeXt-Small model

Loss dynamics related to ConvNeXt-Small model:

Fig 17. (a) is showing the loss dynamics attained by using the ConvNeXt-Small model in our experimental work.

Training VS Value accuracy using ConvNeXt-Small model:

The best validation accuracy of 98.91% was obtained by using the ConvNeXt-Small model in our experiment on the publicly available dataset. In this way it is evident in the fig 17. (b) that our model outperformed very well.

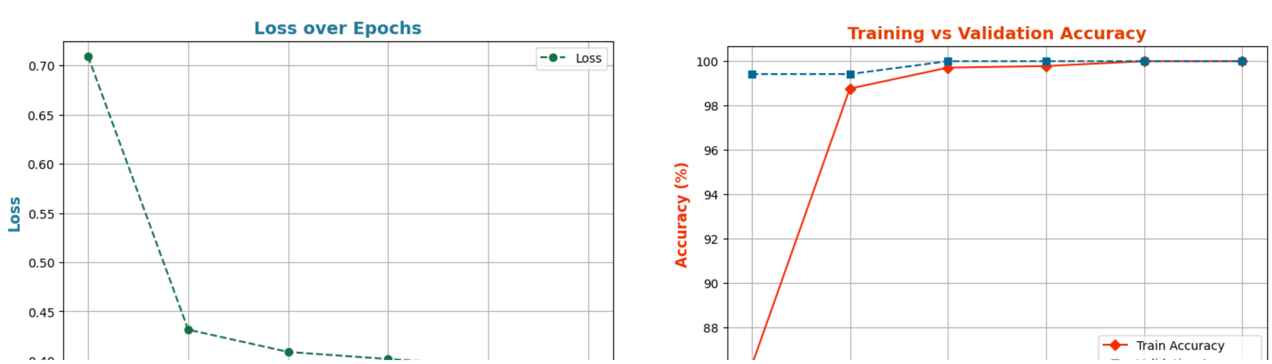


Fig 17. Using ConvNeXt-Small model (a) Showing the loss dynamics (b) Showing the training vs validation accuracy

Comparative view of all the performance measures using different models

In the table 6, we'll be showing the results obtained related to different performance measures such as Accuracy, Precision, Recall, and F1-score using different models as EfficientNet-B0, EfficientNet-B4, ConvNeXt-Base, and ConvNeXt-Small in our work.

Table 6: Showing the performance measures obtained using all the models of our proposed work

Model Name	Accuracy	Precision	F1-Score	Recall
EfficientNet-B0	87.50%	92.13%	81.21%	78.88%
EfficientNet-B4	91.85%	74.61%	76.27%	78.06%
ConvNeXt-Base	96.20%	97.61%	90.02%	87.65%
ConvNeXt-Small	98.91%	99.15%	99.21%	99.30%

Conclusion and future work:

For the fundamental need in sustainable agriculture, it has been considered for application of deep learning models for peanut disease detection. Application of ConvNeXt-Small model were demonstrated to be effective in the identification of all five peanut disease images. The models used in our experimental work showed the better results. This result demonstrates the potential of deep learning to offer farmers effective means of low cost, time appropriate disease detection tools to minimize crop loss and enhance food security. All of this however, leaves little capacity to scale the range of models globally for agricultural applications that see the disease in a variable fashion in a regional manner throughout the environment. The expansion of artificial intelligence driven technology's abilities for environmentally friendly agricultural methods and food safety is what this study provides.

Future research may focus on the application to the presented investigation with advanced deep learning algorithms to further improve the model. Additional information from a wider range of locations globally may increase the model's capability for generalizing well. Deployment of models to edge devices would make the detection of peanut diseases in real time more feasible in the harsh environmental field conditions. Therefore, the inclusion of multimodal data such as spectral or temporal information could help decode in a deeper way, disease progression and improve the diagnostic accuracy.

References:

1. Jha K, Doshi A, Patel P, Shah M. A comprehensive review on automation in agriculture using artificial intelligence. Artificial Intelligence in Agriculture. 2019 Jun 1;2:1–12.
2. Rui M, Ma C, Tang X, Yang J, Jiang F, Pan Y, et al. Phytotoxicity of Silver Nanoparticles to Peanut (Arachis hypogaea L.): Physiological Responses and Food Safety. ACS Sustain Chem Eng [Internet]. 2017 Aug 7 [cited 2025 Apr 23];5(8):6557–67. Available from: https://pubs.acs.org/doi/abs/10.1021/acssuschemeng.7b00736
3. King A. Technology: The Future of Agriculture. Nature 2017 544:7651 [Internet]. 2017 Apr 27 [cited 2025 Apr 23];544(7651):S21–3. Available from: https://www.nature.com/articles/544S21a
4. Anbumozhi A. Leaf Diseases Identification and Classification of Self-Collected Dataset on Groundnut Crop using Progressive Convolutional Neural Network (PGCNN). IJACSA) International Journal of Advanced

Computer Science and Applications [Internet]. 2023 [cited 2025 Apr 23];14(2). Available from: www.ijacsa.thesai.org

5. Efficient Deployment of Peanut Leaf Disease Detection Models on Edge AI Devices - ProQuest [Internet]. [cited 2025 Apr 23]. Available from: https://www.proquest.com/openview/ac4dd7c3017394d3929ca276f20b6d35/1?cbl=2032441&pq-origsite=gscholar

6. Sasmal B, Das A, Dhal KG, Saheb SkB, Khurma RA, Castillo PA. A novel groundnut leaf dataset for detection and classification of groundnut leaf diseases. Data Brief. 2024 Aug;55:110763.

7. Patayon UB, Crisostomo R V. Peanut leaf spot disease identification using pre-trained deep convolutional neural network. International Journal of Electrical and Computer Engineering (IJECE) [Internet]. 2022 Jun 1 [cited 2025 Apr 23];12(3):3005–12. Available from: https://ijece.iaescore.com/index.php/IJECE/article/view/24919

8. Anbumozhi A, Shanthini A. Leaf Diseases Identification and Classification of Self-Collected Dataset on Groundnut Crop using Progressive Convolutional Neural Network (PGCNN). International Journal of Advanced Computer Science and Applications [Internet]. 2023 Dec 28 [cited 2025 Apr 23];14(2):364–73. Available from: www.ijacsa.thesai.org

9. Paramanandham N, Sundhar S, Priya P. Enhancing Disease Detection With Weight Initialization and Residual Connections Using LeafNet for Groundnut Leaf Diseases. IEEE Access. 2024;12:91511–26.

10. M. P. A, Reddy P. Ensemble of CNN models for classification of groundnut plant leaf disease detection. Smart Agricultural Technology. 2023 Dec 1;6:100362.

11. Rakholia RM, Tailor JH, Saini JR, Kaur J, Pahuja H. Groundnuts Leaf Disease Recognition using Neural Network with Progressive Resizing. International Journal of Advanced Computer Science and Applications [Internet]. 2022 Autumn 30 [cited 2025 Apr 24];13(6):83–8. Available from: www.ijacsa.thesai.org

12. Chen SY, Kuo YM, Zhang RH, Cheng TY, Chu PY, Kang LW, et al. The Development of Peanut Defect Detection Technique Using Hyperspectral Images. 2023 Jun 2 [cited 2025 Apr 23]; Available from: https://scispace.com/papers/the-development-of-peanut-defect-detection-technique-using-36bm27t0

13. Wang Y, Ding Z, Song J, Ge Z, Deng Z, Liu Z, et al. Peanut Defect Identification Based on Multispectral Image and Deep Learning. Agronomy 2023, Vol 13, Page 1158 [Internet]. 2023 Apr 19 [cited 2025 Apr 23];13(4):1158. Available from: https://www.mdpi.com/2073-4395/13/4/1158/htm

14. Lv Z, Yang S, Ma S, Wang Q, Sun J, Du L, et al. Efficient Deployment of Peanut Leaf Disease Detection Models on Edge AI Devices. Agriculture 2025, Vol 15, Page 332 [Internet]. 2025 Feb 2 [cited 2025 Apr 23];15(3):332. Available from: https://www.mdpi.com/2077-0472/15/3/332/htm

15. Sivaganesan D. Deep Learning Approaches for Disease Detection in Groundnut Crops using CNN Models. Journal of Soft Computing Paradigm. 2023 Dec;5(4):404–16.

16. Rajmohan M, Reddy DSS, Krishna CM, Krishna NMS. Groundnut leaf disease identification using image processing. AIP Conf Proc [Internet]. 2022 Oct 3 [cited 2025 Apr 23];2519(1). Available from: /aip/acp/article/2519/1/030110/2828013/Groundnut-leaf-disease-identification-using-image

17. Gowrishankar K, Prabha L. An Integrated Image Processing Approach for Diagnosis of Groundnut Plant Leaf Disease using ANN and GLCM. J Sci Ind Res (India). 2020;79:372–6.

18. ZOU Z, CHEN J, WANG L, WU W, YU T, WANG Y, et al. Nondestructive detection of peanuts mildew based on hyperspectral image technology and machine learning algorithm. Food Science and Technology. 2022;42.

19. Tan M, Le Q V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. 36th International Conference on Machine Learning, ICML 2019 [Internet]. 2019 May 28 [cited 2025 Apr 25];2019-June:10691–700. Available from: https://arxiv.org/pdf/1905.11946

20. Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S. A ConvNet for the 2020s. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition [Internet]. 2022 Jan 10 [cited 2025 Apr 24];2022-June:11966–76. Available from: https://arxiv.org/pdf/2201.03545

21. ConvNeXt — Next Generation of Convolutional Networks | by Atakan Erdoğan | Medium [Internet]. [cited 2025 Apr 25]. Available from: https://medium.com/@atakanerdogan305/convnext-next-generation-of-convolutional-networks-325607a08c46